MicroStrategy | 10

System Administration Guide

Tuning, Monitoring, and Troubleshooting your MicroStrategy Business Intelligence System

10.2
Document Number: 09471020

10.2, January 2016

Document Number: 09471020

Copyright © 2016 by MicroStrategy Incorporated. All rights reserved.

If you have not executed a written or electronic agreement with MicroStrategy or any authorized MicroStrategy distributor (any such agreement, a "Separate Agreement"), the following terms apply:

This software and documentation are the proprietary and confidential information of MicroStrategy Incorporated and may not be provided to any other person. Copyright © 2001-2016 by MicroStrategy Incorporated. All rights reserved.

THIS SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" AND WITHOUT EXPRESS OR LIMITED WARRANTY OF ANY KIND BY EITHER MICROSTRATEGY INCORPORATED OR ANYONE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DISTRIBUTION OF THE SOFTWARE OR DOCUMENTATION, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, GOOD TITLE AND NONINFRINGMENT, QUALITY OR ACCURACY. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE AND DOCUMENTATION IS WITH YOU. SHOULD THE SOFTWARE OR DOCUMENTATION PROVE DEFECTIVE, YOU (AND NOT MICROSTRATEGY, INC. OR ANYONE ELSE WHO HAS BEEN INVOLVED WITH THE CREATION, PRODUCTION, OR DISTRIBUTION OF THE SOFTWARE OR DOCUMENTATION) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

In no event will MicroStrategy, Incorporated. or any other person involved with the creation, production, or distribution of the Software be liable to you on account of any claim for damage, including any lost profits, lost savings, or other special, incidental, consequential, or exemplary damages, including but not limited to any damages assessed against or paid by you to any third party, arising from the use, inability to use, quality, or performance of such Software and Documentation, even if MicroStrategy, Inc. or any such other person or entity has been advised of the possibility of such damages, or for the claim by any other party. In addition, MicroStrategy, Inc. or any other person involved in the creation, production, or distribution of the Software shall not be liable for any claim by you or any other party for damages arising from the use, inability to use, quality, or performance of such Software and Documentation, based upon principles of contract warranty, negligence, strict liability for the negligence of indemnity or contribution, the failure of any remedy to achieve its essential purpose, or otherwise. The entire liability of MicroStrategy, Inc. and your exclusive remedy, shall not exceed, at the option of MicroStrategy, Inc., either a full refund of the price paid, or replacement of the Software. No oral or written information given out expands the liability of MicroStrategy, Inc. beyond that specified in the above limitation of liability. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages, so the above limitation may not apply to you.

The information contained in this manual (the Documentation) and the Software are copyrighted and all rights are reserved by MicroStrategy, Inc. MicroStrategy, Inc. reserves the right to make periodic modifications to the Software or the Documentation without obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of the Software or Documentation without prior written consent of an authorized representative of MicroStrategy, Inc. are prohibited. U.S. Government Restricted Rights. It is acknowledged that the Software and Documentation were developed at private expense, that no part is public domain, and that the Software and Documentation are Commercial Computer Software provided with RESTRICTED RIGHTS under Federal Acquisition Regulations and agency supplements to them. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFAR 252.227-7013 et. seq. or subparagraphs (c)(1) and (2) of the Commercial Computer Software-Restricted Rights at FAR 52.227-19, as applicable. Contractor is MicroStrategy, Incorporated., 1850 Towers Crescent Plaza, Tysons Corner, VA 22182. Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.

The following terms and notices apply regardless of whether you have executed a Separate Agreement:

Trademark Information

MicroStrategy, MicroStrategy 10, MicroStrategy 9, MicroStrategy 9s, MicroStrategy Analytics Platform, MicroStrategy Desktop, MicroStrategy Operations Manager, MicroStrategy Analytics Enterprise, MicroStrategy Evaluation Edition, MicroStrategy Secure Enterprise, MicroStrategy Web, MicroStrategy Mobile, MicroStrategy Server, MicroStrategy Parallel Relational In-Memory Engine (MicroStrategy PRIME), MicroStrategy MultiSource, MicroStrategy OLAP Services, MicroStrategy Intelligence Server, MicroStrategy Intelligence Server Universal, MicroStrategy Distribution Services, MicroStrategy Report Services, MicroStrategy Transaction Services, MicroStrategy Visual Insight, MicroStrategy Web Reporter, MicroStrategy Web Analyst, MicroStrategy Office, MicroStrategy Data Mining Services, MicroStrategy Narrowcast Server, MicroStrategy Health Center, MicroStrategy Analyst, MicroStrategy Developer, MicroStrategy Web Professional, MicroStrategy Architect, MicroStrategy SDK, MicroStrategy Command Manager, MicroStrategy Enterprise Manager, MicroStrategy Object Manager, MicroStrategy Integrity Manager, MicroStrategy System Manager, MicroStrategy Analytics App, MicroStrategy Mobile App, MicroStrategy Tech Support App, MicroStrategy Mobile App Platform, MicroStrategy Cloud, MicroStrategy R Integration, Usher, MicroStrategy Usher, Usher Badge, Usher Security, Usher Security Server, Usher Mobile, Usher Analytics, Usher Network Manager, Usher Professional, MicroStrategy Services, MicroStrategy Professional Services, MicroStrategy Consulting, MicroStrategy Customer Services, MicroStrategy Education, MicroStrategy University, MicroStrategy Managed Services, BI QuickStrike, Mobile QuickStrike, Transaction Services QuickStrike Perennial Education Pass, MicroStrategy Web Based Training (WBT), MicroStrategy World, Best in Business Intelligence, Pixel Perfect, Pixel-Perfect, Global Delivery Center, Direct Connect, Enterprise Grade Security For Every Business, Build Your Own Business Apps, Code-Free, Welcome to Ideal, The World's Most Comprehensive Analytics Platform, The World's Most Comprehensive Analytics Platform. Period.

Other product and company names mentioned herein may be the trademarks of their respective owners.

Specifications subject to change without notice. MicroStrategy is not responsible for errors or omissions. MicroStrategy makes no warranties or commitments concerning the availability of future products or versions that may be planned or under development.

Patent Information

This product is patented. One or more of the following patents may apply to the product sold herein: U.S. Patent Nos. 5,321,520, 5,416,602, 5,748,560, 6,154,766, 6,173,310, 6,260,050, 6,263,051, 6,269,393, 6,279,033, 6,501,832, 6,567,796, 6,587,547, 6,606,596, 6,658,093, 6,658,432, 6,662,195, 6,671,715, 6,691,100, 6,694,316, 6,697,808, 6,704,723, 6,707,889, 6,741,980, 6,765,997, 6,768,788, 6,772,137, 6,788,768, 6,792,086, 6,798,867, 6,801,910, 6,820,073, 6,829,334, 6,836,537, 6,850,603, 6,859,798, 6,873,693, 6,885,734, 6,888,929, 6,895,084, 6,940,953, 6,964,012, 6,977,992, 6,996,568, 6,996,569, 7,003,512, 7,010,518, 7,016,480, 7,020,251, 7,039,165, 7,082,422, 7,113,474, 7,113,993, 7,127,403, 7,174,349, 7,181,417, 7,194,457, 7,197,461, 7,228,303, 7,260,577, 7,266,181, 7,272,212, 7,302,639, 7,324,942, 7,330,847, 7,340,040, 7,356,758, 7,356,840, 7,415,438, 7,428,302, 7,430,562, 7,440,898, 7,457,397, 7,486,780, 7,509,671, 7,516,181, 7,559,048, 7,574,376, 7,617,201, 7,725,811, 7,801,967, 7,836,178, 7,861,161, 7,861,253, 7,881,443, 7,925,616, 7,945,584, 7,970,782, 8,005,870, 8,035,382, 8,051,168, 8,051,369, 8,094,788, 8,130,918, 8,296,287, 8,321,411, 8,452,755, 8,521,733, 8,522,192, 8,577,902, 8,606,813, 8,607,138, 8,645,313, 8,761,659, 8,775,807, 8,782,083, 8,812,490, 8,832,588, 8,943,044, 8,943,187. 8,995,628, 9,171,073, 9,116,954, 9,166,986, 9,160,727, 9,172,699, 9,154,303, 9,173,101, 9,154,486, and 9,124,630. Other patent applications are pending.

Third Party Software

Various MicroStrategy products contain the copyrighted technology or software of third parties ("Third Party Software"). A list of Third Party Software, as well as links to any terms and conditions associated with such Third Party Software ("Third Party Terms"), can be found at http://www.microstrategy.com/third-party-notices. Your use of MicroStrategy products is subject to all applicable Third Party Terms.

CONTENTS

Overview and Additional Resources **1**

About this book 4

How to find business scenarios and examples 4

What's new in this guide 5

Who should use this guide 6

Documentation 6

Manuals for MicroStrategy overview and evaluation 8

Resources for Identity and Loyalty 8

Manuals for query, reporting, and analysis 8

Manuals for Analytics Modules 10

Manuals for Narrowcast Services products 10

Software Development Kits 11

Documentation for MicroStrategy Portlets 11

Documentation for MicroStrategy GIS Connectors 11

Help 12

Accessing manuals and other documentation sources 12

Documentation standards 14

© 2016, MicroStrategy Inc.

1. Introduction to MicroStrategy System Administration 1

Best practices for MicroStrategy system administration 2

Understanding the MicroStrategy architecture 3

Storing information: the data warehouse 4

Indexing your data: MicroStrategy metadata 4

Processing your data: Intelligence Server 5

Tying it all together: projects and project sources 6

Communicating with databases 6

Connecting to the MicroStrategy metadata 7

Connecting to the data warehouse 8

Caching database connections 9

Monitoring database instance connections 9

Benefiting from centralized database access control 10

Updating VLDB properties for ODBC connections 11

Managing Intelligence Server 12

What happens when Intelligence Server starts? 12

What happens when Intelligence Server stops? 13

Running Intelligence Server as an application or a service 14

Managing MicroStrategy services from the command line 20

Managing and monitoring projects 24

Managing project status, configuration, or security: Project view 25

Managing clustered Intelligence Servers: Cluster view 26

Setting the status of a project 27

Project and data warehouse maintenance example scenarios 31

Processing jobs 32

Intelligence Server job processing (common to all jobs) 33

Processing report execution 33

Processing object browsing 37

Processing element browsing 38

Processing Report Services document execution 41

© 2016, MicroStrategy Inc.

Processing HTML document execution 42
Client-specific job processing 44
Monitoring currently executing jobs 47
Using automated installation techniques 48
Using a Response file to install the product 48
Using a Response file to configure the product 48
Running a silent installation 48
Security checklist before deploying the system 49
2. Setting Up User Security 51
The MicroStrategy user model 52
About MicroStrategy users 52
About MicroStrategy user groups 52
Privileges 54
Permissions 55
Creating, importing, and deleting users and groups 56
Monitoring users' connections to projects 57
Controlling access to application functionality 58
Controlling access to objects: Permissions 58
Controlling access to functionality: Privileges 66
Defining sets of privileges: Security roles 69
Controlling access to data 73
Controlling access to the database: Connection mappings 74
Linking database users and MicroStrategy users: Passthrough execution 77
Restricting access to data: Security filters 79
Controlling access to data at the database (RDBMS) level 91
Merging users or groups 93
How users and groups are merged 94
Running the User Merge Wizard 96
Workflow: changing authentication modes 97
Modes of authentication 98

© 2016, MicroStrategy Inc.

Configuring the authentication mode for a project source 99
Importing users from different authentication systems 100
Sharing user accounts between users 100
Implementing standard authentication 100
Password policy 101
Steps to implement standard authentication 101
Implementing anonymous authentication 102
Implementing database warehouse authentication 103
Database warehouse authentication information flow 103
Steps to implement database warehouse authentication 104
Database warehouse and metadata (6.x) authentication 105
Implementing LDAP authentication 105
LDAP information flow 106
Checklist: Information required for connecting your LDAP server to MicroStrategy 107
Setting up LDAP authentication in MicroStrategy 123
Managing LDAP authentication 125
Troubleshooting 132
Enabling single sign-on authentication 132
Implementing Windows NT authentication 133
Enabling integrated authentication 140
Enabling Single Sign-on to Web, Mobile, and Office with third-party authentication 161
Enabling Usher authentication for Web and Mobile 177
Authentication examples 180
Security views: Windows authentication and linked warehouse login 180
Connection maps: Standard authentication, connection maps, and partitioned fact tables 181
Overview of SSL and encryption in MicroStrategy 183
Obtaining SSL certificates 184
Prerequisites 185
Generating an SSL Certificate Signing Request 185
Submitting a Certificate Request 186

© 2016, MicroStrategy Inc.

Installing the SSL certificates 187

Adding your enterprise CA as a trusted certificate authority 188

Configuring the MicroStrategy applications to use SSL 190

Configuring SSL between Intelligence Server and Developer, MicroStrategy Web, or Mobile Server 191

Configuring Web, Mobile Server, and Web Services to require SSL access 193

Testing SSL access 194

Configuring MicroStrategy client applications to use an HTTPS URL 195

Enabling encryption between Web or Mobile Server and Intelligence Server 196

Self-signed certificates: Creating a Certificate Authority for development 196

Creating the directories and configuration files for your CA 197

Creating the private key and root certificate for the CA 198

Configuring OpenSSL to use your private key and root certificate 199

Signing certificate requests using your CA 200

5. Managing Your Licenses 203

Managing and verifying your licenses 203

Named User licenses 204

CPU licenses 205

Effects of being out of compliance with your licenses 206

Auditing and updating licenses 206

Using License Manager 207

Auditing your system for the proper licenses 208

Updating your license 209

Updating CPU affinity 210

CPU affinity for Intelligence Server on Windows 211

CPU affinity for Intelligence Server on UNIX/Linux 211

CPU affinity for MicroStrategy Web 214

6. Managing Your Projects 219

The project life cycle 220

Recommended scenario: Development, test, and production 220

© 2016, MicroStrategy Inc.

Real-life scenario: New version from a project developer 222
Implementing the recommended life cycle 223
Duplicating a project 225
What objects are duplicated with a project? 226
Duplicating projects in multiple languages 226
The Project Duplication Wizard 227
Migrating a project to a new database platform 228
Updating projects with new objects 230
Comparing Project Merge to Object Manager 231
Locking projects 231
Copying objects between projects: Object Manager 232
Prerequisites for copying objects between projects 233
Project locking with Object Manager 233
Copying objects 234
What happens when you copy or move an object 237
Resolving conflicts when copying objects 243
Copying objects in a batch: Update packages 249
Merging projects to synchronize objects 266
What happens when you merge projects 267
Merging projects with the Project Merge Wizard 267
Resolving conflicts when merging projects 271
Comparing and tracking projects 272
Comparing objects between two projects 273
Tracking your projects with the Search Export feature 274
Listing unused objects in a project 275
Deleting unused schema objects: managed objects 276
Deleting managed objects one-by-one 276
Deleting all unused managed objects 277
7. Monitoring System Usage 279
MicroStrategy system monitors 279

© 2016, MicroStrategy Inc.

Monitoring system activity: Change journaling 281
Enabling change journaling 281
Viewing the change journal entries 283
Increasing the number of change journal entries to view or export 284
Searching the change journal for relevant entries 285
Exporting the change journal 286
Purging the change journal 286
Monitoring system usage: Intelligence Server statistics 288
Overview of Intelligence Server statistics 288
Best practices for recording Intelligence Server statistics 292
Configuring Intelligence Server to log statistics 293
Additional monitoring tools 299
Diagnostics and Performance Logging Tool 299
Integrity Manager 299
Enterprise Manager 299
Health Center 299
Windows Performance Monitor 300
8. Tuning Your System for Best Performance 301
Tuning overview and best practices 302
Defining the system requirements 302
Configuring the system design 303
Best practices for tuning your system 305
Designing system architecture 306
How the data warehouse can affect performance 306
How the network can affect performance 307
How clustering can affect performance 310
Managing system resources 310
Managing system memory and resources: Windows Performance Monitor 311
Processor type, speed, and number of processors 311
Physical disk 312

© 2016, MicroStrategy Inc.

Memory 312
Governing Intelligence Server memory use with Memory Contract Manager 316
Governing memory for requests from MicroStrategy Web products 323
Managing user sessions 324
Governing concurrent users 325
Governing user resources 327
Governing user profiles 329
Governing requests 331
Excluding document datasets from the job limits 332
Limiting the total number of jobs 332
Limiting the number of jobs per project 333
Limiting the number of jobs per user session and per user account 333
Limiting the number of executing jobs per user and project 334
Limiting the maximum report execution time 334
Limiting a report's SQL per pass 335
Managing job execution 336
Managing database connection threads 337
Prioritizing jobs 340
Results processing 343
Governing results delivery 347
Limiting the information displayed at one time 348
Limiting the number of XML cells 349
Limiting export sizes 350
Limiting the memory consumption for file generation 350
Limiting the total number of XML drill paths 351
Tuning your system for in-memory datasets 352
Configuring Intelligence Server for in-memory datasets 352
Configuring your projects for in-memory datasets 352
Designing reports 353
Analytic complexity 353

© 2016, MicroStrategy Inc.

Subtotals 354
Page-by feature 354
Prompt complexity 354
Report Services documents 354
Intelligent Cubes 355
Configuring Intelligence Server and projects 355
Intelligence Server Configuration Editor 356
Project Configuration Editor 361
Database connection 365
VLDB settings 366
Tuning Narrowcast Server and Intelligence Server 367
Application design considerations 367
How Narrowcast Server connects to Intelligence Server 368
9. Clustering Multiple MicroStrategy Servers 369
Overview of clustering 370
Benefits of clustering 370
The clustered architecture 372
Synchronizing cached information across nodes in a cluster 373
Prerequisites for clustering Intelligence Servers 377
MicroStrategy prerequisites 377
Server prerequisites 378
Prerequisites for Windows clustering 379
Prerequisites for UNIX/Linux clustering 379
Clustering Intelligence Servers 380
Configuring caches in a cluster 380
Joining the nodes in a cluster 386
Verifying the clustered system is working 387
Distributing projects across nodes in a cluster 388
Managing your clustered system 390
Managing your projects across nodes of a cluster 390

© 2016, MicroStrategy Inc.

Project failover and latency 392
Shutting down a node 394
Maintaining result caches and History Lists in a clustered environment 395
Connecting MicroStrategy Web to a cluster 397
Clustering and firewalls 397
Exporting to PDF or Excel 397
Node failure 397
10. Improving Report and Document Response Time: Caching 399
Result caches 400
Cache management best practices 402
Types of result caches 403
Location of result caches 404
Cache matching algorithm 406
Disabling result caching 408
Monitoring result caches 409
Managing result caches 412
Configuring result cache settings 416
Saving report results: History List 425
Understanding History Lists 426
Configuring History List data storage 429
Accessing History Lists 432
Archiving History List messages 434
Managing History Lists 435
Element caches 440
Element caching terminology 441
Location of element caches 442
Cache matching algorithm 442
Enabling or disabling element caching 442
Limiting the number of elements displayed and cached at a time 443
Caching algorithm 446

© 2016, MicroStrategy Inc.

Limiting the amount of memory available for element caches 446
Limiting which attribute elements a user can see 448
Limiting element caches by database connection 449
Limiting element caches by database login 449
Deleting all element caches 450
Summary table of element cache settings 451
Object caches 451
Cache matching algorithm 452
Enabling or disabling object caching 452
Limiting the amount of memory available for object caches 452
Deleting object caches 454
Summary table of object caching settings 455
11. Managing Intelligent Cubes 457
Managing Intelligent Cubes: Intelligent Cube Monitor 458
Viewing Intelligent Cube information and usage statistics 458
Monitoring and modifying Intelligent Cube status 460
Governing Intelligent Cube memory usage, loading, and storage 464
Governing Intelligent Cube memory usage 465
Loading Intelligent Cubes when Intelligence Server starts 471
Storing Intelligent Cubes in secondary storage 473
Supporting connection mappings in Intelligent Cubes 474
12. Scheduling Jobs and Administrative Tasks 477
Best practices for scheduling jobs and administrative tasks 478
Creating and managing schedules 479
Time-triggered schedules 480
Event-triggered schedules 480
Creating schedules 480
Managing schedules 481
About events and event-triggered schedules 483
Scheduling administrative tasks 485

© 2016, MicroStrategy Inc.

Managing scheduled administration tasks 487
Scheduling reports and documents: Subscriptions 488
Types of subscriptions 489
Creating subscriptions 490
Managing subscriptions 498
Configuring and administering Distribution Services 501
Overview of Distribution Services 501
Best practices for using Distribution Services 504
High-level checklist to set up a report delivery system 505
Creating and managing transmitters 506
Creating and managing devices 512
Creating and managing contacts 522
13. Administering MicroStrategy Web and Mobile 529
Assigning privileges for MicroStrategy Web 530
Using the MicroStrategy Web Administrator page 531
Controlling access to the Administrator page 531
Defining project defaults 533
Loading and applying default values 533
Using additional security features for MicroStrategy Web and Mobile 534
Using firewalls 534
Using cookies 537
Using encryption 538
Applying file-level security 539
Sample MicroStrategy system 540
Integrating Narrowcast Server with MicroStrategy Web products 541
Enabling users to install MicroStrategy Office from Web 542
FAQs for configuring and tuning MicroStrategy Web products 542
How do I configure my MicroStrategy Web environment if I have a user community of x users? How much hardware am I going to need? 542
How do time-out settings in MicroStrategy Web and Intelligence Server affect MicroStrategy Web users? 543

© 2016, MicroStrategy Inc.

How can I tune my MicroStrategy Web server for best performance? 544

14. Combining Administrative Tasks with System Manager 545

Creating a workflow 546

Using connectors to create the logical order of a workflow 553

Using entry processes to determine the first step in a workflow 554

Using decisions to determine the next step in a workflow 555

Using exit processes to end a workflow 559

Processing related tasks one by one 560

Processing multiple tasks simultaneously 562

Using comments to provide context and information to a workflow 567

Validating a workflow 568

Supporting loops in a workflow to attempt configurations multiple times 570

Sample workflows: Templates 571

Defining processes 578

Configuring MicroStrategy components 579

Managing projects 582

Administering Intelligence Servers and other MicroStrategy services 586

Automating administrative tasks 589

Verifying reports and documents 592

Creating data source names 595

Completing a separate System Manager workflow 607

Retrieving MicroStrategy information 609

Performing system processes 609

Administering cloud-based environments 625

Determining process resolution using exit codes 637

Using parameters for processes 638

Using wildcard characters in processes 643

Deploying a workflow 644

Using System Manager to test and deploy a workflow 644

Using the command line to deploy a workflow 647

© 2016, MicroStrategy Inc.

15. Automating Administrative Tasks with Command Manager 651

Using Command Manager 651

Privileges required for using Command Manager 652

Creating and executing scripts 653

Procedures in Command Manager 654

Executing a Command Manager script 656

Encrypting Command Manager scripts 659

Project and configuration locking 659

Handling syntax errors 660

Handling execution errors 660

Timeout errors 661

Command Manager script syntax 662

Using Command Manager from the command line 663

Using Command Manager with OEM software 663

16. Verifying Reports and Documents with Integrity Manager 665

What is an integrity test? 666

Types of integrity tests 667

Testing Intelligence Server performance 668

Best practices for using Integrity Manager 669

Creating an integrity test 670

Saving and loading a test 671

Executing an integrity test 672

Executing a test from the command line 673

Executing a test against a remote Intelligence Server 676

Executing prompted reports with Integrity Manager 676

Executing a test under multiple MicroStrategy user accounts 680

Ignoring dynamic SQL when comparing SQL 681

Matching equivalent SQL strings 682

Viewing the results of a test 683

Viewing and editing notes 684

© 2016, MicroStrategy Inc.

Grouping differences in graph and PDF reports 685
Accessing the saved results of a test 686
List of tags in the integrity test file 688
17. Maintaining Your MicroStrategy System with Health Center 699
Accessing the Health Center console 700
Configuring and managing a Health Center system 700
Best practices for configuring a Health Center system 701
The Master Health Agent 702
Health Agents 703
Adding Health Agents to the system 704
Diagnosing and fixing problems 708
Fixing problems found by Health Center 709
Analyzing log files 710
Collecting and sending information for MicroStrategy Technical Support 711
Keeping Health Center up to date 712
Manually updating your Health Center system 713
List of system checks 713
System checks for Configuration 715
System checks for Developer 715
System checks for Enterprise Manager 716
System checks for Intelligence Server 716
System checks for Narrowcast Server 720
System checks for MicroStrategy Web products 721
Using a response file to configure a Health Agent or Master Health Agent 722
Creating a response file 723
Executing a response file 723
Health Center response file parameters 725
18. Troubleshooting 729
Methodology for finding trouble spots 730
Intelligence Server stopped 730

© 2016, MicroStrategy Inc.

The system is slow 730
Users cannot log in 731
Finding trouble spots using diagnostics 731
Configuring what is logged 732
Viewing and analyzing log files 745
Analyzing a server state dump 747
Memory depletion troubleshooting 750
Basic memory concepts 751
Potential causes of memory depletion 752
Authentication troubleshooting 756
Troubleshooting Windows authentication in MicroStrategy Web 756
Failure to log in to server (three-tier) project source 756
Troubleshooting LDAP authentication 757
Fixing inconsistencies in the metadata 762
Fixing deleted items 762
Object dependencies troubleshooting 765
Date/time functions troubleshooting 765
Performance troubleshooting 766
Project performance 766
Schema update performance 767
Cache Monitor and Intelligent Cube Monitor performance 767
Troubleshooting data imported from a file 767
Subscription and report results troubleshooting 768
Drilled-to report returns no data or incorrect data 768
Graph report results 769
Number of report result rows 769
Subscription with a prompt is not sent and disappears 769
Contact does not appear in list of contacts when subscribing 769
SQL/MDX string length 770
Freeform SQL report error 771

© 2016, MicroStrategy Inc.

Internationalization troubleshooting 771

Object displays translations in a language different from the project or object default language 772

User's preferred language defaults to project default language 772

Custom names for system objects are not retained after upgrade 772

Troubleshooting Intelligence Server 772

Logon failure 772

Failure to activate Intelligence Server 773

Multiple instances of Intelligence Server 774

Clustered environments troubleshooting 774

Problems in a clustered environment 774

What order should I start the server nodes? 774

Which node am I connected to? 775

Do I need multiple project sources? 775

Does it matter which node I connect to in Developer? 775

Can I control which node I connect to in MicroStrategy Web? 775

Caches in a clustered environment 775

Node synchronization in a clustered environment 776

Statistics logging troubleshooting 777

Statistics are not being logged 777

Statistics on subscriptions and deliveries 779

Additional resources for troubleshooting 780

MicroStrategy Readme and release notes 780

MicroStrategy Knowledge Base 781

Customer Forums 782

Technical Support 782

Glossary 783

Index 795

© 2016, MicroStrategy Inc.

© 2016, MicroStrategy Inc.

OVERVIEW AND ADDITIONAL RESOURCES

Description of this guide

This guide is to be the primary resource that system administrators use to learn about the concepts and high-level steps for implementing, deploying, maintaining, tuning, and troubleshooting the MicroStrategy business intelligence system. It offers a full discussion of the concepts that a system administrator should consider before the system is made widely available to users in the enterprise.

For additional reference information about seldom-used or advanced topics such as VLDB properties, internationalization, and the Intelligence Server statistics data dictionaries, see the Supplemental Reference for System Administration.

The chapters provide the following information:

- *Introduction to MicroStrategy System Administration*

  This chapter provides an overview of the architecture and how the MicroStrategy system interacts with the various external components/systems. It describes how Intelligence Server connects to and uses the data warehouse. It also describes what Intelligence Server is, what happens when it is started and stopped, and what MicroStrategy metadata is and what purposes it serves, as well as what a MicroStrategy project is and what MicroStrategy objects are. It describes all aspects of connecting to databases including database instances, database connections, and

© 2016, MicroStrategy Inc.

what a MicroStrategy server definition is and what it controls. It also describes general job processing flows with the MicroStrategy system including report execution, object and element browsing, and HTML document execution.

- *Chapter 2, Setting Up User Security*

  This chapter covers what users and groups are, what the different modes are for authentication and how to implement them, how to control access to data at both the application and database levels, and how to control access to the application functionality. The examples section shows how combinations of security features in both the MicroStrategy system and in the database management systems can be used together.

  This chapter describes how to manage MicroStrategy Web and MicroStrategy Web Universal, what the Web-related privileges are for the product, how to use the Administrator page including how to set project defaults. It also describes additional security requirements or options you can use with MicroStrategy Web products, including using digital certificates or firewalls, secure sockets layers, and so on.

- *Chapter , Identifying Users: Authentication*

  Authentication is the process through which the system identifies the user. This chapter describes the modes of authentication that MicroStrategy supports, and how to configure them so that they support your user community.

- *Chapter , Enabling Secure Communication*

  This chapter describes the steps to enable secure, encrypted communications between MicroStrategy components using SSL.

- *Chapter 5, Managing Your Licenses*

  This chapter covers making the system available to users. This includes some best practices are for deploying the system, and how to implement easy ways to install systems using SMS systems and silent installs; what License Manager is and how to use it; and setting up security in the MicroStrategy environment.

- *Chapter 6, Managing Your Projects*

  In a MicroStrategy system, a project is the environment in which reporting is done. This chapter provides information on how to manage a project's life cycle, how to duplicate a project, update or copy project objects, merge projects, compare and track projects, and manage schema objects.

- *Chapter 7, Monitoring System Usage*

  This chapter explains how you can use the monitors available in the system to see the state of the system at any time (past or present). It describes how Enterprise Manager can help do this by monitoring statistics that can be logged.

- *Chapter 8, Tuning Your System for Best Performance*

  This chapter provides information for you to find the balance that maximizes the use of your system's capacity to provide the best performance possible for the required number of users.

- *Chapter 9, Clustering Multiple MicroStrategy Servers*

© 2016, MicroStrategy Inc.

A clustered set of machines provides a related set of functionality or services to a common set of users. MicroStrategy recommends clustering Intelligence Servers in environments where access to the data warehouse is mission-critical and system performance is of utmost importance. This chapter describes how to cluster Intelligence Servers, how to manage clustered projects, and how to connect MicroStrategy Web to a cluster.

- *Chapter 10, Improving Report and Document Response Time: Caching*

  This chapter explains how you can make the system efficient and remove load from Intelligence Server by using the caching and History List features. It describes how caches work in the system, where they are stored, what the matching requirements are for using a cache, how to create pre-calculated data using aggregate tables, how to administer caches including how to invalidate them. It also describes what the History List is, how it is used in both MicroStrategy Web and Developer, and how to administer it.

- *Chapter 11, Managing Intelligent Cubes*

  You can return data from your data warehouse and save it to Intelligence Server memory, rather than directly displaying the results in a report. This data can then be shared as a single in-memory copy, among many different reports created by multiple users. The reports created from the shared sets of data are executed against the in-memory copy, also known as an Intelligent Cube. This chapter provides details to understand and to create Intelligent Cubes your users can access when the execute reports and documents.

- *Chapter 12, Scheduling Jobs and Administrative Tasks*

  This chapter describes how you can automate certain MicroStrategy jobs and administrative tasks. Methods of automation include scheduling reports, documents, and administrative tasks, and using MicroStrategy Distribution Services to distribute reports and documents via email, file, and printer subscriptions.

- *Chapter 13, Administering MicroStrategy Web and Mobile*

  This chapter provides a high-level overview for some of the administrative tasks that are unique to administering MicroStrategy Web, Web Universal, and Mobile Server.

- *Chapter 14, Combining Administrative Tasks with System Manager*

  System Manager lets you define multiple configurations for your MicroStrategy environment, that can then be executed in a single workflow. This provides the ability to deploy the various configurations to as many systems as required. The deployment of these configurations can be done using a standard interface, an interactive command line process, or a completely silent configuration process.

- *Chapter 15, Automating Administrative Tasks with Command Manager*

  Command Manager lets you automate various administrative and application development tasks by using text commands that can be saved as scripts. This chapter describes how to create and execute these scripts.

- *Chapter 16, Verifying Reports and Documents with Integrity Manager*

© 2016, MicroStrategy Inc.

Integrity Manager is an automated comparison tool designed to streamline the testing of MicroStrategy reports and documents. It can verify that changes to the environment have not caused changes to the report results, and can also test the performance of an Intelligence Server. This chapter describes how to configure Integrity Manager and how to create and execute an integrity test, and provides best practices for using Integrity Manager.

- *Chapter 17, Maintaining Your MicroStrategy System with Health Center*

  MicroStrategy Health Center can help you diagnose and fix problems in your MicroStrategy system. It detects known problems and provides an immediate solution to many of them. This chapter describes how to configure your Health Center network and how to schedule system checks for MicroStrategy components.

- *Chapter 18, Troubleshooting*

  This chapter provides a high-level methodology for finding trouble spots in the system and fixing them. It describes how to use the Diagnostics and Performance Logging tool to help diagnose bottlenecks in the system, memory depletions, exceptions, or authentication problems.

About this book

The following sections provide the location of additional examples and describe the user roles for which the information in this book was designed.

The sample documents and images in this guide, as well as some example steps, were created with dates that may no longer be available in the MicroStrategy Tutorial project. If you are re-creating an example, replace the year(s) shown in this guide with the most recent year(s) available in the software.

How to find business scenarios and examples

Within this guide, many of the concepts discussed are accompanied by business scenarios or other descriptive examples.

For examples of reporting functionality, see the MicroStrategy Tutorial, which is MicroStrategy's sample warehouse and project. Information about the MicroStrategy Tutorial and the Human Resources Analytics Module can be found in the Basic Reporting Guide.

Detailed examples of advanced reporting functionality can be found in the Advanced Reporting Guide.

Other examples in this book use the Analytics Module project, which includes a set of precreated sample reports. Sample reports present data for analysis in the human resources business area.

 © 2016, MicroStrategy Inc.

What's new in this guide

MicroStrategy 10

- You can allow users of MicroStrategy Mobile to use integrated authentication to log into MicroStrategy. For steps to configure integrated authentication, see *Enabling integrated authentication, page 140*.
- System Manager:
  - Encrypt or decrypt specified text or a file (see *Encrypting/Decrypting text or files, page 610*).
  - Determine the state of Amazon Cloud image (see *Getting cloud image status, page 627*).

MicroStrategy Analytics Enterprise

- The name of MicroStrategy Desktop has been changed to MicroStrategy Developer.

MicroStrategy 9.4

- System Manager:
  - Retrieve information from sources including a folder, the contents of a file, or a System Manager parameter. This information can then be passed to another process in the System Manager workflow for processing a task (see *Processing related tasks one by one, page 560*).
  - Create a file or folder as part of a System Manager workflow with the new Rename Files process (see *Creating a file or folder, page 612*).
  - Determine the number of files in a folder as part of a System Manager workflow with the new Count File process (see *Determining the number of files in a folder, page 612*).
  - Rename a file or folder as part of a System Manager workflow with the new Rename Files process (see *Renaming a file or folder, page 616*).
  - Provide a network name and determine how IP addresses are assigned when creating a new virtual machine (see *Creating a virtual machine, page 635*).
- When implementing LDAP authentication, it is recommended that you use the LDAP Connectivity Wizard. For instructions, see *Implementing LDAP authentication, page 105*.
- Link users' integrated authentication IDs to their LDAP user and group information. For steps, see *Linking integrated authentication users to LDAP users, page 142*.

© 2016, MicroStrategy Inc.

Who should use this guide

This document is designed for:

- System administrators responsible for configuring and maintaining the MicroStrategy business intelligence system
- Database administrators who may need to understand how databases (such as the data warehouse and metadata) work with the MicroStrategy system
- Network administrators who may need to configure network connections between the system's components

Education

MicroStrategy Education Services provides a comprehensive curriculum and highly skilled education consultants. Many customers and partners from over 800 different organizations have benefited from MicroStrategy instruction.

Courses that can help you prepare for using this manual or that address some of the information in this manual include:

- Implementing MicroStrategy: Development and Deployment
- MicroStrategy Administration

For the most up-to-date and detailed description of education offerings and course curricula, visit *http://www.microstrategy.com/Education*.

Documentation

MicroStrategy provides both manuals and online help; these two information sources provide different types of information, as described below:

- **Manuals**: In general, MicroStrategy manuals provide:
  - Introductory information and concepts
  - Examples and images
  - Checklists and high-level procedures to get started

  The steps to access the manuals are described in *Accessing manuals and other documentation sources, page 12*.

  Most of these manuals are also available printed in a bound, soft cover format. To purchase printed manuals, contact your MicroStrategy Account Executive with a purchase order number.
- **Help**: In general, MicroStrategy help provides:

 © 2016, MicroStrategy Inc.

- Detailed steps to perform procedures
- Descriptions of each option on every software screen

Translations

For the most up-to-date translations of MicroStrategy documentation, refer to the MicroStrategy Knowledge Base. Due to translation time, manuals in languages other than English may contain information that is one or more releases behind. You can see the version number on the title page of each manual.

Finding information

You can search all MicroStrategy books and Help for a word or phrase, with a simple Google™ search at http://www.google.com. For example, type "MicroStrategy derived metric" or "MicroStrategy logical table" into a Google search. As described above, books typically describe general concepts and examples; Help typically provides detailed steps and screen options. To limit your search to MicroStrategy books, on Google's main page you can click **More**, then select **Books**.

Additional formats

MicroStrategy manuals are available as electronic publications, downloadable on the Apple iBookstore or Google Play, and can be read on your iOS or Android device respectively. To download a book, search for the book's title in the iBookstore or Google Play respectively. To view a list of manuals that are currently available, scan the following QR codes using your device's camera:

For iOS devices, scan the following QR code:

For Android devices, scan the following QR code:

For new MicroStrategy releases, it may take several days for the latest manuals to be available on the iBookstore or Google Play.

Manuals for MicroStrategy overview and evaluation

- *Introduction to MicroStrategy: Evaluation Guide*

  Instructions for installing, configuring, and using the MicroStrategy Evaluation Edition of the software. This guide also includes a detailed, step-by-step evaluation process of MicroStrategy features, where you perform reporting with the MicroStrategy Tutorial project and its sample business data.

- *MicroStrategy Evaluation Edition Quick Start Guide*

  Overview of the installation and evaluation process, and additional resources.

- *MicroStrategy Suite: Quick Start Guide*

  Evaluate MicroStrategy as a departmental solution. Provides detailed information to download, install, configure, and use the MicroStrategy Suite.

Resources for Identity and Loyalty

- *Alert Commerce Management System (CMS) Guide* and *Alert API Reference*

  Content resources providing steps to deliver and manage marketing and commerce content through the Alert mobile applications.

- *Usher Help*

  Steps to perform mobile identity validation using the Usher mobile identity network to issue electronic badges for identifying users.

Manuals for query, reporting, and analysis

- *MicroStrategy Installation and Configuration Guide*

  Information to install and configure MicroStrategy products on Windows, UNIX, Linux, and HP platforms, as well as basic maintenance guidelines.

- *MicroStrategy Upgrade Guide*

  Instructions to upgrade existing MicroStrategy products.

- *MicroStrategy Project Design Guide*

  Information to create and modify MicroStrategy projects, and understand facts, attributes, hierarchies, transformations, advanced schemas, and project optimization.

 © 2016, MicroStrategy Inc.

- *MicroStrategy Basic Reporting Guide*

  Instructions to get started with MicroStrategy Developer and MicroStrategy Web, and how to analyze data in a report. Includes the basics for creating reports, metrics, filters, and prompts.

- *MicroStrategy Advanced Reporting Guide: Enhancing Your Business Intelligence Application*

  Instructions for advanced topics in the MicroStrategy system, building on information in the Basic Reporting Guide. Topics include reports, Freeform SQL reports, Query Builder reports, filters, metrics, Data Mining Services, custom groups, consolidations, and prompts.

- *MicroStrategy Report Services Document Creation Guide: Creating Boardroom Quality Documents*

  Instructions to design and create Report Services documents, building on information in the *Document and Dashboard Analysis Guide*. It is organized to help guide you through creating a new document, from creating the document itself, to adding objects to the new document, and formatting the document and its objects.

- *MicroStrategy Dashboards and Widgets Creation Guide: Creating Interactive Dashboards for your Data*

  Instructions for designing and creating MicroStrategy Report Services dashboards, a type of document that is optimized for viewing online and for user interactivity. It builds on the basic concepts about documents presented in the *MicroStrategy Report Services Document Creation Guide*.

- *MicroStrategy OLAP Services Guide*

  Information on MicroStrategy OLAP Services, which is an extension of MicroStrategy Intelligence Server. OLAP Services features include Intelligent Cubes, derived metrics, derived elements, dynamic aggregation, view filters, and dynamic sourcing.

- *MicroStrategy Office User Guide*

  Instructions for using MicroStrategy Office to work with MicroStrategy reports and documents in Microsoft® Excel, PowerPoint, and Word, to analyze, format, and distribute business data.

- *MicroStrategy Mobile Analysis Guide: Analyzing Data with MicroStrategy Mobile*

  Information and instructions for using MicroStrategy Mobile to view and analyze data, and perform other business tasks with MicroStrategy reports and documents on a mobile device.

- *MicroStrategy Mobile Design and Administration Guide: A Platform for Mobile Intelligence*

© 2016, MicroStrategy Inc.

Information and instructions to install and configure MicroStrategy Mobile, as well as instructions for a designer working in MicroStrategy Developer or MicroStrategy Web to create effective reports and documents for use with MicroStrategy Mobile.

- *MicroStrategy System Administration Guide: Tuning, Monitoring, and Troubleshooting your MicroStrategy Business Intelligence System*

  Concepts and high-level steps to implement, deploy, maintain, tune, and troubleshoot a MicroStrategy business intelligence system.

- *MicroStrategy Supplemental Reference for System Administration: VLDB Properties, Internationalization, User Privileges, and other Supplemental Information for Administrators*

  Information and instructions for MicroStrategy administrative tasks such as configuring VLDB properties and defining data and metadata internationalization, and reference material for other administrative tasks.

- *MicroStrategy Functions Reference*

  Function syntax and formula components; instructions to use functions in metrics, filters, attribute forms; examples of functions in business scenarios.

- *MicroStrategy MDX Cube Reporting Guide*

  Information to integrate MicroStrategy with MDX cube sources. You can integrate data from MDX cube sources into your MicroStrategy projects and applications.

Manuals for Analytics Modules

- *Analytics Modules Installation and Porting Guide*
- *Customer Analysis Module Reference*
- *Sales Force Analysis Module Reference*
- *Financial Reporting Analysis Module Reference*
- *Sales and Distribution Analysis Module Reference*
- *Human Resources Analysis Module Reference*

Manuals for Narrowcast Services products

- *MicroStrategy Narrowcast Server Getting Started Guide*

  Instructions to work with the tutorial to learn Narrowcast Server interfaces and features.

- *MicroStrategy Narrowcast Server Installation and Configuration Guide*

  Information to install and configure Narrowcast Server.

 © 2016, MicroStrategy Inc.

- *MicroStrategy Narrowcast Server Application Designer Guide*

  Fundamentals of designing Narrowcast Server applications.

- *MicroStrategy Narrowcast Server System Administrator Guide*

  Concepts and high-level steps to implement, maintain, tune, and troubleshoot Narrowcast Server.

- *MicroStrategy Narrowcast Server Upgrade Guide*

  Instructions to upgrade an existing Narrowcast Server.

Software Development Kits

- *MicroStrategy Developer Library (MSDL)*

  Information to understand the MicroStrategy SDK, including details about architecture, object models, customization scenarios, code samples, and so on.

- *MicroStrategy Web SDK*

  The Web SDK is available in the MicroStrategy Developer Library, which is part of the MicroStrategy SDK.

- *Narrowcast Server SDK Guide*

  Instructions to customize Narrowcast Server functionality, integrate Narrowcast Server with other systems, and embed Narrowcast Server functionality within other applications. Documents the Narrowcast Server Delivery Engine and Subscription Portal APIs, and the Narrowcast Server SPI.

Documentation for MicroStrategy Portlets

- *Enterprise Portal Integration Help*

  Information to help you implement and deploy MicroStrategy BI within your enterprise portal, including instructions for installing and configuring out-of-the-box MicroStrategy Portlets for several major enterprise portal servers.

  This resource can be accessed from the MicroStrategy Product Manuals page, as described in *Accessing manuals and other documentation sources, page 12*.

Documentation for MicroStrategy GIS Connectors

- *GIS Integration Help*

Information to help you integrate MicroStrategy with Geospatial Information Systems (GIS), including specific examples for integrating with various third-party mapping services.

This resource can be accessed from the MicroStrategy Product Manuals page, as described in *Accessing manuals and other documentation sources, page 12*.

Help

Each MicroStrategy product includes an integrated help system to complement the various interfaces of the product as well as the tasks that can be accomplished using the product.

Some of the MicroStrategy help systems require a web browser to be viewed. For supported web browsers, see the MicroStrategy Readme.

MicroStrategy provides several ways to access help:

- Help button: Use the Help button or ? (question mark) icon on most software windows to see help for that window.
- Help menu: From the Help menu or link at the top of any screen, select MicroStrategy Help to see the table of contents, the Search field, and the index for the help system.
- F1 key: Press F1 to see context-sensitive help that describes each option in the software window you are currently viewing.

For MicroStrategy Web, MicroStrategy Web Administrator, and MicroStrategy Mobile Server, pressing the F1 key opens the context-sensitive help for the web browser you are using to access these MicroStrategy interfaces. Use the Help menu or ? (question mark) icon to access help for these MicroStrategy interfaces.

Accessing manuals and other documentation sources

The manuals are available from http://www.microstrategy.com/producthelp, as well as from the machine where MicroStrategy was installed.

Adobe Acrobat Reader is required to view these manuals. If you do not have Acrobat Reader installed on your computer, you can download it from http://get.adobe.com/reader/.

The best place for all users to begin is with the *MicroStrategy Basic Reporting Guide*.

To access the installed manuals and other documentation sources, see the following procedures:

- *To access documentation resources from any location, page 13*
- *To access documentation resources on Windows, page 13*

 © 2016, MicroStrategy Inc.

- *To access documentation resources on UNIX and Linux , page 13*

To access documentation resources from any location

1 Visit http://www.microstrategy.com/producthelp.

To access documentation resources on Windows

1 From the Windows **Start** menu, choose **Programs** (or **All Programs**), **MicroStrategy Documentation**, then **Product Manuals**. A page opens in your browser showing a list of available manuals in PDF format and other documentation sources.

2 Click the link for the desired manual or other documentation source.

3 If you click the link for the Narrowcast Services SDK Guide, a File Download dialog box opens. This documentation resource must be downloaded. Select **Open this file from its current location**, and click **OK**.

If bookmarks are not visible on the left side of an Acrobat (PDF) manual, from the **View** menu click **Bookmarks and Page**. This step varies slightly depending on your version of Adobe Acrobat Reader.

To access documentation resources on UNIX and Linux

1 Within your UNIX or Linux machine, navigate to the directory where you installed MicroStrategy. The default location is `/opt/MicroStrategy`, or `$HOME/MicroStrategy/install` if you do not have write access to `/opt/MicroStrategy`.

2 From the MicroStrategy installation directory, open the `Help` folder.

3 Open the `Product_Manuals.htm` file in a web browser. A page opens in your browser showing a list of available manuals in PDF format and other documentation sources.

4 Click the link for the desired manual or other documentation source.

5 If you click the link for the Narrowcast Services SDK Guide, a File Download dialog box opens. This documentation resource must be downloaded. Select **Open this file from its current location**, and click **OK**.

If bookmarks are not visible on the left side of an Acrobat (PDF) manual, from the **View** menu click **Bookmarks and Page**. This step varies slightly depending on your version of Adobe Acrobat Reader.

Documentation standards

MicroStrategy online help and PDF manuals (available both online and in printed format) use standards to help you identify certain types of content. The following table lists these standards.

These standards may differ depending on the language of this manual; some languages have rules that supersede the table below.

Type	Indicates
bold	• Button names, check boxes, options, lists, and menus that are the focus of actions or part of a list of such GUI elements and their definitions Example: Click **Select Warehouse**.
italic	• Names of other product manuals and documentation resources • When part of a command syntax, indicates variable information to be replaced by the user Example: The *aggregation level* is the level of calculation for the metric. Example: Type `copy c:\`*filename* `d:\`*foldername*`\`*filename*
`Courier font`	• Calculations • Code samples • Registry keys • Path and file names • URLs • Messages displayed in the screen • Text to be entered by the user Example: `Sum(revenue)/number of months.` Example: Type `cmdmgr -f scriptfile.scp` and press **Enter**.
+	A keyboard command that calls for the use of more than one key (for example, SHIFT+F1).
	A note icon indicates helpful information for specific situations.
	A warning icon alerts you to important information such as potential security risks; these should be read before continuing.

 © 2016, MicroStrategy Inc.

1

INTRODUCTION TO MICROSTRATEGY SYSTEM ADMINISTRATION

Introduction

This chapter summarizes the major components in the MicroStrategy system architecture and provides a brief overview of some of the basic concepts you need to understand to administer a MicroStrategy system.

The following are discussed:

- *Best practices for MicroStrategy system administration*
- *Understanding the MicroStrategy architecture, page 3*
- *Communicating with databases, page 6*
- *Managing Intelligence Server, page 12*
- *Managing and monitoring projects, page 24*
- *Processing jobs, page 32*
- *Using automated installation techniques, page 48*
- *Security checklist before deploying the system, page 49*

© 2016, MicroStrategy Inc.

Best practices for MicroStrategy system administration

MicroStrategy recommends the following best practices to keep your system running smoothly and efficiently:

- Use the project life cycle of *development, testing, production* to fully test your reports, metrics, and other objects before releasing them to users. For an in-depth explanation of the project life cycle, see *The project life cycle, page 220*.
- If you need to delegate administrative responsibilities among several people, you can create separate security roles for each type of administrator and assign those roles to the appropriate users. MicroStrategy comes with a number of predefined administrative security roles for this purpose. For more information about these security roles, see *The role-based administration model, page 72*. For an introduction to the MicroStrategy security model, including users and privileges, see *Chapter 2, Setting Up User Security*.
- Once Intelligence Server is up and running, you can adjust its governing settings to better suit your environment. For detailed information about these settings, see *Chapter 8, Tuning Your System for Best Performance*.

  You can use Enterprise Manager to monitor various aspects of Intelligence Server's performance. Enterprise Manager is a MicroStrategy project that uses the Intelligence Server statistics database as its data warehouse. For information on Enterprise Manager, see the Enterprise Manager chapter in the *MicroStrategy Operations Manager Guide*.
- If you have multiple machines available to run Intelligence Server, you can cluster those machines to improve performance and reliability. For detailed information about clustering, including instructions, see *Chapter 9, Clustering Multiple MicroStrategy Servers*.
- Create caches for commonly used reports and documents to reduce the database load and improve the system response time. For information about caching, including instructions on how to set up caching, see *Chapter 10, Improving Report and Document Response Time: Caching*.

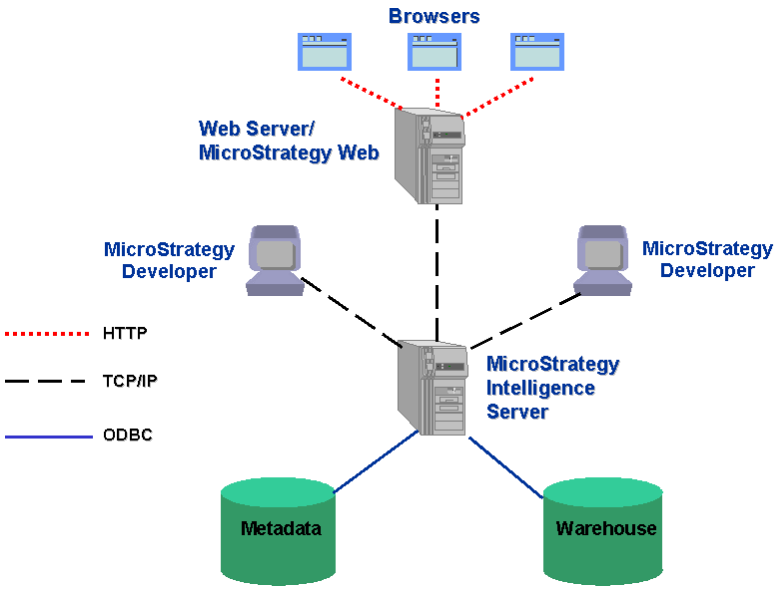
  Creating reports based on Intelligent Cubes can also greatly speed up the processing time for reports. Intelligent Cubes are part of the OLAP Services features in Intelligence Server. For an overview of Intelligent Cubes, see *Chapter 11, Managing Intelligent Cubes*.
- Schedule administrative tasks and reports to run during off-peak hours, so that they do not adversely affect system performance. For instructions on scheduling reports and administrative tasks, see *Chapter 12, Scheduling Jobs and Administrative Tasks*.

© 2016, MicroStrategy Inc.

You can automate the delivery of reports and documents to users with the Distribution Services add-on to Intelligence Server. For an overview of Distribution Services, see *Overview of Distribution Services, page 501*.

Understanding the MicroStrategy architecture

A MicroStrategy system is built around a three-tier or four-tier structure. The diagram below illustrates a four-tier system.



- The first tier, at the bottom, consists of two databases: the data warehouse, which contains the information that your users analyze; and the MicroStrategy metadata, which contains information about your MicroStrategy projects. For an introduction to these databases, see *Storing information: the data warehouse* and *Indexing your data: MicroStrategy metadata*.
- The second tier consists of MicroStrategy Intelligence Server, which executes your reports against the data warehouse. For an introduction to Intelligence Server, see *Processing your data: Intelligence Server*.

If MicroStrategy Developer users connect via a two-tier project source (also called a direct connection), they can access the data warehouse without Intelligence Server. For more information on two-tier project sources, see *Tying it all together: projects and project sources*.

- The third tier in this system is MicroStrategy Web or Mobile Server, which delivers the reports to a client. For an introduction to MicroStrategy Web, see *Chapter 13, Administering MicroStrategy Web and Mobile*.

© 2016, MicroStrategy Inc.

- The last tier is the MicroStrategy Web client or MicroStrategy Mobile app, which provides documents and reports to the users.

In a three-tier system, Developer is the last tier.

For a more complete discussion of business intelligence architecture, see the first chapter of the *MicroStrategy Project Design Guide*.

Storing information: the data warehouse

The data warehouse is the foundation that your MicroStrategy system is built on. It stores all the information you and your users analyze with the MicroStrategy system. This information is usually placed or loaded in the data warehouse using some sort of extraction, transformation, and loading (ETL) process. Your online transaction processing (OLTP) system is usually the main source of original data that the ETL process uses.

As a system administrator, you need to know which relational database management system (RDBMS) manages your data warehouse, how the MicroStrategy system accesses it (which machine it is on and which ODBC driver and Data Source Name it uses to connect to it), and what should happen when the data warehouse is loaded (such as running scripts to invalidate certain caches in Intelligence Server, and so on). These are all discussed later in this guide.

Indexing your data: MicroStrategy metadata

MicroStrategy metadata is like a road map or an index to the information that is stored in your data warehouse. The MicroStrategy system uses the metadata to know where in the data warehouse it should look for information. It also stores other types of objects that allow you to access that information. These are discussed below.

The metadata resides in a database, the *metadata repository*, that is separate from your data warehouse. This can be initially created when you run through the MicroStrategy Configuration Wizard. All the metadata information is encrypted and stored in database tables defined by MicroStrategy.

For more information about running the MicroStrategy Configuration Wizard, see the Installation and Configuration Guide.

To help explain how the MicroStrategy system uses the metadata to do its work, imagine that a user runs a report with a total of revenue for a certain region in a quarter of the year. The metadata stores information about how the revenue metric is to be calculated, information about which rows and tables in the data warehouse to use for the region, and the most efficient way to retrieve the information.

The physical warehouse schema is a type of conceptual tool that is crucial for you to visualize information's location in the data warehouse. This includes table and column information about where things are actually stored as well as maps, such as lookup and relate tables, that help the system efficiently access that information. Persons who create the schema objects in the MicroStrategy metadata must reference the physical

 © 2016, MicroStrategy Inc.

warehouse schema. Therefore, it is not actually stored in a location in the metadata, but it is implicitly present in the definition of the schema objects in the metadata.

The role of the physical warehouse schema is further explained in the Basic Reporting Guide.

In addition to the physical warehouse schema's implicit presence in the metadata, the following types of objects are stored in the metadata:

- Schema objects are objects created, usually by a project designer or architect, based on the logical and physical models. Facts, attributes, and hierarchies are examples of schema objects. These objects are developed in MicroStrategy Architect, which can be accessed from MicroStrategy Developer. The Project Design Guide is devoted to explaining schema objects.
- Application objects are the objects that are necessary to run reports. These objects are generally created by a report designer and can include reports, report templates, filters, metrics, prompts, and so on. These objects are built in Developer or Command Manager. The Basic Reporting Guide and Advanced Reporting Guide are devoted to explaining application objects.
- Configuration objects are administrative and connectivity-related objects. They are managed in Developer (or Command Manager) by an administrator changing the Intelligence Server configuration or project configuration. Examples of configuration objects include users, groups, server definitions and so on. Configuration objects are primarily covered in the System Administration Guide.

Processing your data: Intelligence Server

Intelligence Server is the second tier in the MicroStrategy system. It must be running for users to get information from your data warehouse using MicroStrategy clients such as MicroStrategy Web or Developer.

Intelligence Server is the heart of the MicroStrategy system. It executes reports stored in the metadata against the data warehouse and passes on the results of those reports to the users. For detailed information about Intelligence Server, including how to start and stop it, see *Managing Intelligence Server, page 12*.

A server definition is an instance of Intelligence Server and all its configuration settings. Multiple server definitions can be stored in the metadata, but only one can be run at a time on a machine. If you want multiple machines to point to the same metadata, you should cluster them. For more information about clustering, including instructions on how to cluster Intelligence Servers, see *Chapter 9, Clustering Multiple MicroStrategy Servers*.

Pointing multiple Intelligence Servers to the same metadata without clustering may cause metadata inconsistencies. This configuration is not supported, and MicroStrategy strongly recommends that users not configure their systems in this way.

The server definition information includes the following:

- Metadata connectivity information, such as:

    - Metadata DSN
    - Metadata ID and encrypted password
- MicroStrategy administrator user name
- Intelligence Server configuration settings—set in Developer

Tying it all together: projects and project sources

A MicroStrategy project is an object in which you define all the schema and application objects, which together provide for a flexible reporting environment. A project's metadata repository is established by the project source in which you construct the project. The project's data warehouse is specified by associating the project with the appropriate database instance. For detailed information about projects, including instructions on how to create a project, see the Project Design Guide.

You can manage your projects using the System Administration Monitor. For details, see *Managing and monitoring projects, page 24*.

A project source is a container stored in Developer that defines how Developer accesses the metadata repository. Think of a project source as a pointer to one or more projects that are stored in a metadata repository.

Two types of project sources can be created, defined by the type of connection they represent:

- **Server** connection, or three-tier, which specifies the Intelligence Server to connect to.
- **Direct** connection, or two-tier, which bypasses Intelligence Server and allows Developer to connect directly to the MicroStrategy metadata and data warehouse. Note that this is primarily for project design and testing. Because this type of connection bypasses Intelligence Server, important benefits such as caching and governing, which help protect the system from being overloaded, are not available.

In older systems you may encounter a **6.x Project** connection (also two-tier) that connects directly to a MicroStrategy version 6 project in read-only mode.

For more information on project sources, see the Installation and Configuration Guide.

Communicating with databases

To successfully configure your system, you must establish connections to the MicroStrategy metadata and to the data warehouse that contains the business information on which you will report. These procedures are explained in the Installation and Configuration Guide.

 © 2016, MicroStrategy Inc.

While the MicroStrategy Configuration Wizard sets up some of these connections for you automatically when you first install and configure your MicroStrategy software, you may need to further fine-tune them. For instructions on how to manage your database connections, see *Monitoring database instance connections, page 9*.

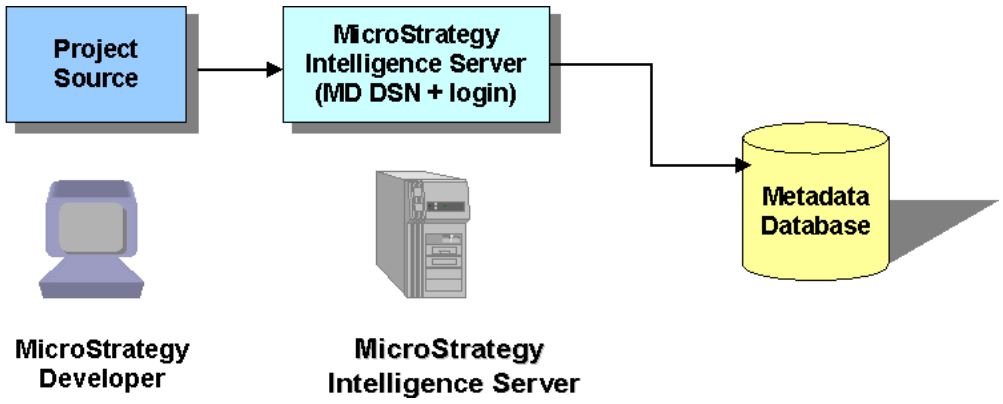
Connecting to the MicroStrategy metadata

MicroStrategy users need connectivity to the metadata so that they can access projects, create objects, and execute reports. Intelligence Server connects to the metadata by reading the server definition registry when it starts. However, this connection is only one segment of the connectivity picture.

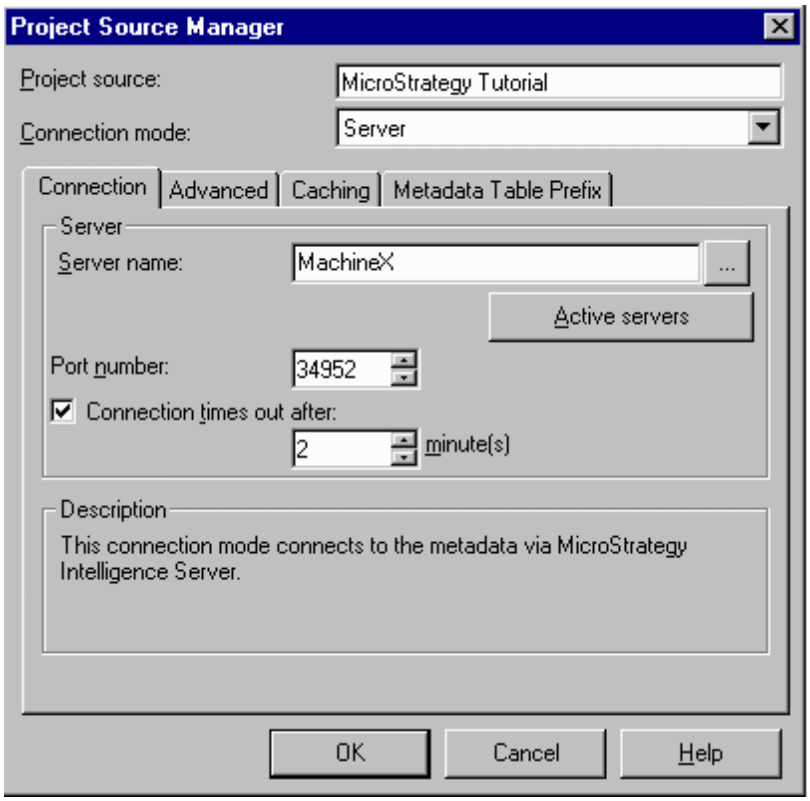
Consider these questions:

- How does a Developer user access the metadata?
- How does a user connect to Intelligence Server?
- Where is the connection information stored?

The diagram below illustrates three-tier metadata connectivity between the MicroStrategy metadata database (tier one), Intelligence Server (tier two), and Developer (tier three).



In a server (three-tier) environment, Developer metadata connectivity is established through the project source. For steps to create a project source, see the Installation and Configuration Guide.



You can also create and edit a project source using the Project Source Manager in Developer. When you use the Project Source Manager, you must specify the Intelligence Server machine to which to connect. It is through this connection that Developer users retrieve metadata information.

The Developer connection information is stored in the Developer machine registry.

Connecting to the data warehouse

Once you establish a connection to the metadata, you must create a connection to the data warehouse. This is generally performed during initial software installation and configuration, but it can also be established with the following procedures in Developer:

- Creating a database instance: A MicroStrategy object created in Developer that represents a connection to the data warehouse. A database instance specifies warehouse connection information such as the data warehouse DSN, Login ID and password, and other data warehouse-specific information.
- Creating a database connection: Specifies the DSN and database login used to access the data warehouse. A database instance designates one database connection as the default connection for MicroStrategy users.
- Creating a database login: Specifies the user ID and password used to access the data warehouse. The database login overwrites any login information stored in the DSN.

 © 2016, MicroStrategy Inc.

- User connection mapping: The process of mapping MicroStrategy users to database connections and database logins. To execute reports, MicroStrategy users must be mapped to a database connection and login.

For procedures to connect to the data warehouse, see the Installation and Configuration Guide.

Caching database connections

Connecting to and disconnecting from databases incurs a small amount of overhead that may cause a small yet noticeable decrease in performance in high-concurrency systems. With connection caching, Intelligence Server is able to reuse database connections. This minimizes the overhead associated with repeatedly connecting to and disconnecting from databases.

Connections can exist in one of two states:

- Busy: connections that are actively submitting a query to a database
- Cached: connections that are still connected to a database but not actively submitting a query to a database

A cached connection is used for a job if the following criteria are satisfied:

- The connection string for the cached connection matches the connection string that will be used for the job.
- The driver mode (multiprocess versus multithreaded) for the cached connection matches the driver mode that will be used for the job.

Intelligence Server does not cache any connections that have pre- or post-SQL statements associated with them because these options may drastically alter the state of the connection.

Monitoring database instance connections

A warehouse database connection is initiated any time a user executes an uncached report or browses uncached elements. The Database Connection Monitor enables you to view the number of busy and cached connections to the data warehouse. You can also view the name of the database instance, the user who is using the connection, and the database login being used to connect to the database.

If a database connection is cached, the ODBC connection from Intelligence Server to the data warehouse remains open. However, if the data warehouse connection surpasses the connection time-out or lifetime governors (set in the Database Connections dialog box, on the **Advanced** tab), the ODBC connection closes, and it no longer displays in the Database Connection Monitor.

© 2016, MicroStrategy Inc.

To view the current database connections

1. In Developer, log in to a project source. You must log in as a user with the Monitor Database Connections privilege.
2. Expand **Administration**, then expand **System Monitors**, and then select **Database Connections**. The database connection information displays on the right-hand side.

To delete a database connection

In the Database Connection Monitor, right-click the connection and select **Disconnect**.

Benefiting from centralized database access control

All database connectivity is handled by Intelligence Server, which provides centralized control of database access. The advantages of centralized control include:

- Connectionless client—All connections to databases in the system are made through Intelligence Server. This means that only the Intelligence Server machine needs to have database connectivity. It also eliminates the need to rely on identically configured connections on client and server computers. This makes it easy to set up, deploy, and manage large systems.
- Connection caching—Connecting to and disconnecting from databases incurs a small amount of overhead that may cause a small, yet noticeable, decrease in performance in high-concurrency systems. With connection caching, Intelligence Server is able to reuse database connections. This minimizes the overhead associated with repeated connecting to and disconnecting from databases.
- Workload governing—Because only Intelligence Server connects to databases, it can make sure that no one database becomes overloaded with user requests. This is especially important for the data warehouse.
- User connection mapping—Intelligence Server can map MicroStrategy users and user groups to data warehouse login IDs. This allows multiple users to access the database using a single database login.
- Ease of administration/monitoring—Because all database connectivity is handled by Intelligence Server, keeping track of all connections to all databases in the system is easy.
- Prioritized access to databases—You can set access priority by user, project, estimated job cost, or any combination of these.
- Multiprocess execution—The ability to run in multiprocess mode means that if one process fails, such as a lost or hung database access thread, the others are not affected.

 © 2016, MicroStrategy Inc.

- Database optimizations—Using VLDB properties, Intelligence Server is able to take advantage of the unique performance optimizations that different database servers offer.

Updating VLDB properties for ODBC connections

VLDB properties allow Intelligence Server to take advantage of the unique optimizations that different databases offer. Depending on the database type, these properties can affect how Intelligence Server handles things like:

- Join options, such as the star join and full outer join
- Metric calculation options, such as when to check for NULLs and zeros
- Pre- and post-SQL statements
- Query optimizations, such as sub-queries and driving tables
- Table types, such as temporary tables or derived tables

For more information about all the VLDB properties, see the *VLDB Properties* chapter in the Supplemental Reference for System Administration.

Upgrading your database type properties

Default VLDB properties are set according to the database type specified in the database instance. MicroStrategy periodically updates the default settings as database vendors add new functionality.

When you create the metadata for a MicroStrategy project, the database-specific information is loaded from a file supplied by MicroStrategy (called `Database.pds`). If you get a new release from MicroStrategy, the metadata is automatically upgraded using the `Database.pds` file with the metadata update process. The Administrator is the only user who can upgrade the metadata. Do this by clicking **Yes** when prompted for updating the metadata. This happens when you connect to an existing project after installing a new MicroStrategy release.

The MicroStrategy system cannot detect when you upgrade or change the database used to store the MicroStrategy metadata or your data warehouse. If you upgrade or change the database that is used to store the metadata or data warehouse, you can manually update the database type to apply the default properties for the new database type.

When the you update the database type information, this process:

- Loads newly supported database types. For example, properties for the newest database servers that were recently added.
- Loads updated properties for existing database types that are still supported.
- Keeps properties for existing database types that are no longer supported. If there were no updates for an existing database type, but the properties for it have been

removed from the `Database.pds` file, the process does not remove them from your metadata.

In some cases, MicroStrategy no longer updates certain DBMS objects as newer versions are released. These are not normally removed. However, in the case of Oracle 8i R2 and Oracle 8i R3, the DBMS objects were merged into "Oracle 8i R2/R3" for both Standard and Enterprise editions because Oracle 8i R3 is no longer being updated. You may need to select the merged version as part of your database instance if you are using a version of Oracle 8i. This will become apparent if date/time functions stop working, particularly in Enterprise Manager.

For more information about VLDB properties, see the *VLDB Properties* chapter in the Supplemental Reference for System Administration.

You may need to manually upgrade the database types if you chose not to run the update metadata process after installing a new release.

To manually upgrade the database type properties

1. In the Database Instance editor, click the **General** tab.
2. Select **Upgrade**. The Upgrade Database Type dialog box opens.

For more detailed information about manually upgrading VLDB properties, functions, and SQL syntax for your database server, see the *MicroStrategy Developer Help*.

The *MicroStrategy Readme* lists all DBMSs that are supported or certified for use with MicroStrategy.

Managing Intelligence Server

This section introduces you to basic Intelligence Server operation, including starting and stopping Intelligence Server and running it as a service or as an application.

You can improve your system and database performance by adjusting various Intelligence Server governing settings to fit your system parameters and your reporting needs. For detailed information about these settings, see *Chapter 8, Tuning Your System for Best Performance*.

What happens when Intelligence Server starts?

Once a server definition is defined and selected for Intelligence Server using the Configuration Wizard, the metadata connection information specified in the server

 © 2016, MicroStrategy Inc.

definition is saved in the machine's registry. When Intelligence Server starts, it reads this information to identify the metadata to which it will connect.

The portion of server definition information that is stored in the machine's registry includes the server definition name, the DSN pointing to the metadata, and the metadata ID and encrypted password. For more information on server definitions, see the Installation and Configuration Guide.

When Intelligence Server starts, it does the following:

- Initializes internal processing units
- Reads from the machine registry which server definition it is supposed to use and connects to the specified metadata database
- Loads configuration and schema information for each loaded project
- Loads existing report cache files from automatic backup files into memory for each loaded project (up to the specified maximum RAM setting)

  This occurs only if report caching is enabled and the Load caches on startup feature is enabled.

- Loads schedules
- Loads MDX cube schemas

  You can set Intelligence Server to load MDX cube schemas when it starts, rather than loading MDX cube schemas upon running an MDX cube report. For more details on this and steps to load MDX cube schemas when Intelligence Server starts, see the *Configuring and Connecting Intelligence Server* chapter of the Installation and Configuration Guide.

If a system or power failure occurs, Intelligence Server cannot capture its current state. The next time the server is started, it loads the state information, caches, and History Lists that were saved in the last automatic backup. (The automatic backup frequency is set using the Intelligence Server Configuration Editor.) The server does not re-execute any job that was running until the person requesting the job logs in again.

What happens when Intelligence Server stops?

When you initiate an Intelligence Server shutdown, it:

- Writes cache and History List information to backup files
- Cancels currently executing jobs

  The user who submitted a canceled job sees a message in the History List indicating that there was an error. The user must resubmit the job.

- Closes database connections
- Logs out connected users from the system

- Removes itself from the cluster (if it was in a cluster)

It does not rejoin the cluster automatically when restarted.

As noted earlier, if a system or power failure occurs, these actions cannot be done. Instead, Intelligence Server recovers its state from the latest automatic backup.

Running Intelligence Server as an application or a service

Intelligence Server can be started as a Windows service or as an application. If you run Intelligence Server as a service, you can start and stop it from a remote machine with Developer or by logging into the Intelligence Server machine remotely. In addition, you can configure the service to start automatically when the machine on which it is installed starts. For more information about running Intelligence Server as a service, see *Starting and stopping Intelligence Server as a service, page 15*.

On rare occasions you may need to run Intelligence Server as an application. This includes occasions when you need precise control over when Intelligence Server stops and starts or when you need to change certain advanced tuning settings that are not available when Intelligence Server is running as a service. For more information about running Intelligence Server as an application, see *Starting Intelligence Server as an application, page 19*.

Registering and unregistering Intelligence Server as a UNIX service

In UNIX, when you configure Intelligence Server you must specify that it starts as an application or a service. If you want to start Intelligence Server as a service, you must register it as a service with the system. In addition, in UNIX, if you want to start Intelligence Server as a service after having started it as an application, you must register it as a service.

To register or unregister Intelligence Server as a service in UNIX, you must be logged in to the Intelligence Server machine with root privileges.

You can register Intelligence Server as a service in two ways:

- From the Configuration Wizard: on the Specify a Port Number page, ensure that the **Register Intelligence Server as a Service** check box is selected.
- From the command line: in `~/MicroStrategy/bin` enter:

  ```
  mstrctl -s IntelligenceServer rs
  ```

If you want to start Intelligence Server as an application after having registered it as a service, you need to unregister it. Unregistering the service can be done only from the command line, in `~/MicroStrategy/bin`. The syntax to unregister the service is:

```
mstrctl -s IntelligenceServer us
```

 © 2016, MicroStrategy Inc.

Starting and stopping Intelligence Server as a service

Once the service is started, it is designed to run constantly, even after the user who started it logs off the system. However, you may need to stop and restart it for these reasons:

- Routine maintenance on the Intelligence Server machine
- Changes to Intelligence Server configuration options that cannot be changed while Intelligence Server is running
- Potential power outages due to storms or planned building maintenance

You can start and stop Intelligence Server manually as a service using any of the following methods:

- MicroStrategy Service Manager is a management application that can run in the background on the Intelligence Server machine. It is often the most convenient way to start and stop Intelligence Server. For instructions, see *Service Manager, page 15*.
- If you are already using Developer, you may need to start and stop Intelligence Server from within Developer. For instructions, see *Developer, page 18*.
- Use the web-based monitoring tool, MicroStrategy Operations Manager, to stop and start local and remote Intelligence Servers. For an overview, see *Operations Manager, page 18*
- You can start and stop Intelligence Server as part of a Command Manager script. For details, see *Command Manager, page 18*.
- Finally, you can start and stop Intelligence Server from the command line using MicroStrategy Server Control Utility. For instructions, see *Command line, page 18*.

Prerequisites for starting and stopping Intelligence Server

- You must have the Configuration access permission for the server definition object. For information about object permissions in MicroStrategy, see *Controlling access to objects: Permissions, page 58*. For a list of the permission groupings for server definition objects, see *Permissions for server governing and configuration, page 62*.
- To remotely start and stop the Intelligence Server service in Windows, you must be logged in to the remote machine as a Windows user with administrative privileges.

Service Manager

Service Manager is a management tool installed with Intelligence Server that enables you to start and stop Intelligence Server and choose a startup option. For instructions on how to use Service Manager, click **Help** from within Service Manager.

Service Manager requires that port 8888 be open. If this port is not open, contact your network administrator.

To open MicroStrategy Service Manager in Windows

1. In the system tray of the Windows task bar, double-click the MicroStrategy Service Manager icon, or .

2. If the icon is not present in the system tray, then from the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, then select **Service Manager**.

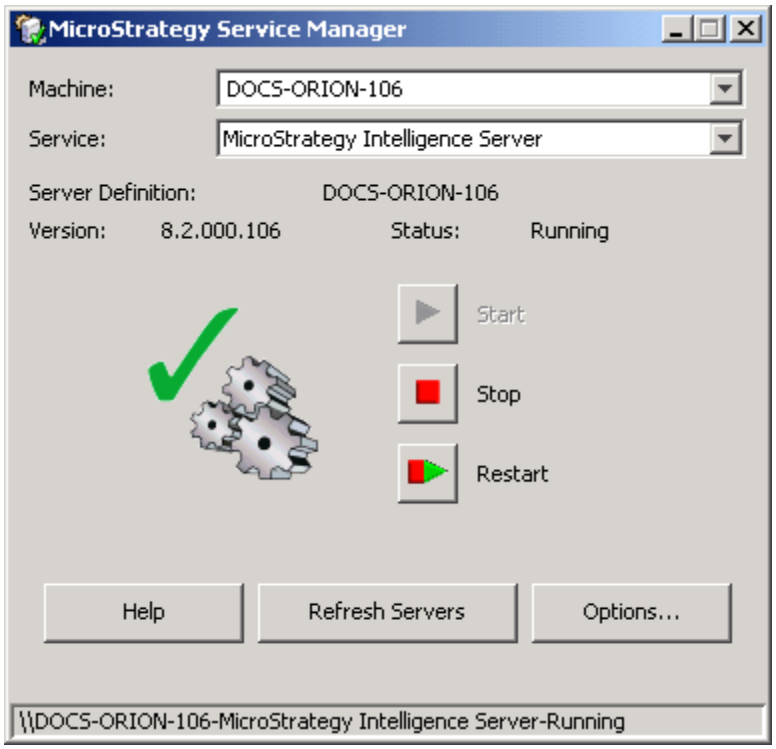
   The MicroStrategy Service Manager dialog box opens.

To open MicroStrategy Service Manager in UNIX

In UNIX, Service Manager requires an X-Windows environment.

1. Browse to the folder specified as the home directory during MicroStrategy installation (the default is `~/MicroStrategy`), then browse to `/bin`.

2. Type `./mstrsvcmgr` and press **ENTER**. The MicroStrategy Service Manager dialog box opens.



 © 2016, MicroStrategy Inc.

Using the Listener/Restarter to start Intelligence Server

You can configure Intelligence Server to start automatically when the Intelligence Server machine starts. You can also configure the Restarter to restart the Intelligence Server service automatically if it fails, but the machine on which it is installed is still running. To do this, you must have the MicroStrategy Listener service running.

To start a MicroStrategy service automatically when the machine restarts

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, then select **Service Manager**. The MicroStrategy Service Manager dialog box opens.
2. In the **Server** drop-down list, select the name of the machine on which the service is installed.
3. In the **Service** drop-down list, select the service.
4. Click **Options.** The Service Options dialog box opens.
5. Select **Automatic** as the **Startup Type** option.
6. Click **OK**.

You can also set this using the Services option in the Microsoft Window's Control Panel.

To start Intelligence Server service automatically when it fails unexpectedly

The MicroStrategy Listener service must be running for the Re-starter feature to work.

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, then select **Service Manager**. The MicroStrategy Service Manager dialog box opens.
2. In the **Server** drop-down list, select the machine on which the Intelligence Server service is installed.
3. In the **Service** drop-down list, select **MicroStrategy Intelligence Server**.
4. Click **Options.** The Service Options dialog box opens.
5. On the Intelligence Server Options tab, select the **Enabled** check box for the **Re-starter Option**.

Developer

You can start and stop a local Intelligence Server from Developer. You cannot start or stop a remote Intelligence Server from Developer; you must use one of the other methods to start or stop a remote Intelligence Server.

To start or stop Intelligence Server using Developer

1. In Developer, in the Folder List, right-click the **Administration** icon.
2. Choose **Start Server** to start it or **Stop Server** to stop it.

Operations Manager

Operations Manager is a web-based console used to manage and monitor your MicroStrategy system. You can use Operations Manager to start and stop local and remote Intelligence Servers in your environment. For steps, see the *MicroStrategy Operations Manager Guide*.

Command Manager

Command Manager is a script-based tool that enables you to perform various administrative and maintenance tasks with reusable scripts. You can start and stop Intelligence Server using Command Manager.

For the Command Manager syntax for starting and stopping Intelligence Server, see the Command Manager Help (press **F1** from within Command Manager). For a more general introduction to MicroStrategy Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*.

Command line

You can start and stop Intelligence Server from a command prompt, using the MicroStrategy Server Control Utility. This utility is invoked by the command `mstrctl`. By default the utility is in `C:\Program Files (x86)\Common Files\MicroStrategy\` in Windows, and in `~/MicroStrategy/bin` in UNIX.

The syntax to start the service is:

```
mstrctl -s IntelligenceServer start --service
```

The syntax to stop the service is:

```
mstrctl -s IntelligenceServer stop
```

For detailed instructions on how to use the Server Control Utility, see *Managing MicroStrategy services from the command line, page 20*.

 © 2016, MicroStrategy Inc.

Windows Services window

You can start and stop Intelligence Server and choose a startup option using the Windows Services window.

To start and stop Intelligence Server using the Windows Services window

1. On the Windows **Start** menu, point to **Settings**, then choose **Control Panel**. The Control Panel window opens.
2. Double-click **Administrative Tools**, and then double-click **Services**. The Services window opens.
3. From the **Services** list, select **MicroStrategy Intelligence Server**. The Properties dialog box for the Intelligence Server service opens.
4. You can do any of the following:
   - To start the service, click **Start**.
   - To stop the service, click **Stop**.
   - To change the startup type, select a startup option from the drop-down list.
     - **Automatic** means that the service starts when the computer starts.
     - **Manual** means that you must start the service manually.
     - **Disabled** means that you cannot start the service until you change the startup type to one of the other types.
5. When you are finished, click **OK** to close the Properties dialog box.

Starting Intelligence Server as an application

While the need to do so is rare, you can start Intelligence Server as an application. This may be necessary if you must administer Intelligence Server on the machine on which it is installed, if Developer is not installed on that machine.

Some advanced tuning settings are only available when starting Intelligence Server as a service. If you change these settings, they are applied the next time Intelligence Server is started as a service.

MicroStrategy recommends that you not change these settings unless requested to do so by a MicroStrategy Technical Support associate.

There are some limitations to running Intelligence Server as an application:

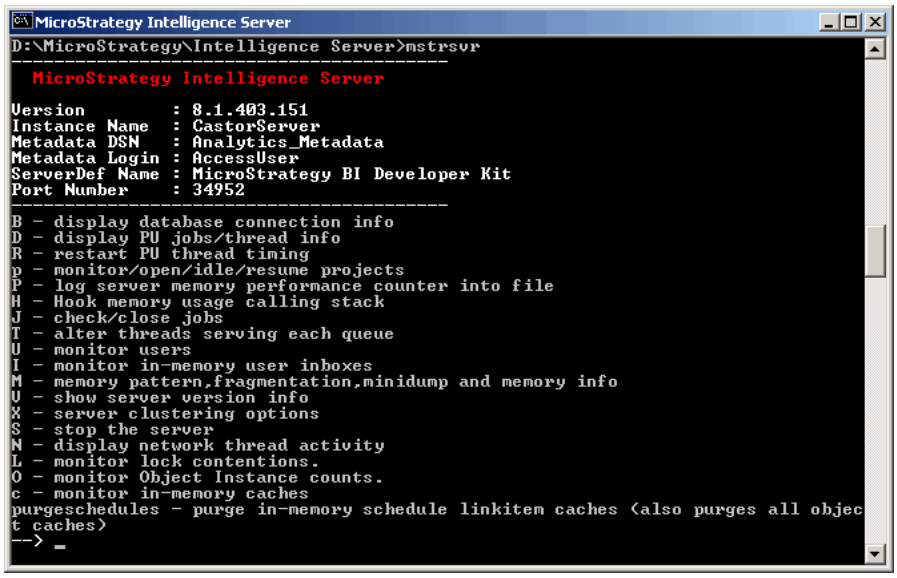
- The user who starts Intelligence Server as an application must remain logged on to the machine for Intelligence Server to keep running. When the user logs off, Intelligence Server stops.

- If Intelligence Server is started as an application, you cannot administer it remotely. You can administer it only by logging in to the Intelligence Server machine.
- The application does not automatically restart if it fails.

In UNIX, if Intelligence Server has previously been configured to run as a service, you must unregister it as a service before you can run it as an application. For instructions on unregistering Intelligence Server as a service, see *Registering and unregistering Intelligence Server as a UNIX service, page 14*.

The default path for the Intelligence Server application executable is `C:\Program Files (x86)\MicroStrategy\Intelligence Server\MSTRSvr.exe` in Windows, and `~/MicroStrategy/bin` in UNIX.

Executing this file from the command line displays the following administration menu in Windows, and a similar menu in UNIX.



To use these options, type the corresponding letter on the command line and press **Enter**. For example, to monitor users, type `U` and press **Enter**. The information is displayed.

Managing MicroStrategy services from the command line

MicroStrategy Server Control Utility enables you to create and manage Intelligence Server server instances from the command line. A server instance is an Intelligence Server that is using a particular server definition. For more information about server definitions, see *Processing your data: Intelligence Server*.

Server Control Utility can also be used to start, stop, and restart other MicroStrategy services—such as the Listener, Distribution Manager, Execution Engine, or Enterprise

 © 2016, MicroStrategy Inc.

Manager Data Loader services—and to view and set configuration information for those services.

The following table lists the commands that you can perform with the Server Control Utility. The syntax for using the Server Control Utility commands is:

```
mstrctl -m machinename [-l login] -s servicename
command [instancename]
[(> | <) filename.xml]
```

where:

- *machinename* is the name of the machine hosting the server instance or service. If this parameter is omitted, the service is assumed to be hosted on the local machine.
- *login* is the login for the machine hosting the server instance or service, and is required if you are not logged into that machine. You are prompted for a password.
- *servicename* is the name of the service, such as IntelligenceServer or EMService.

To retrieve a list of services on a machine, use the command `mstrctl -m machinename ls`.

- *command* is one of the commands from the list below.
- *instancename* is the name of a server instance, where required. If a name is not specified, the command uses the default instance name.
- *filename* is the name of the file to read from or write to.

If you want to. . .	Then use this command. . .
Get information about the Server Control Utility	
List all commands for the Server Control Utility. **Note**: This command does not require a machine name, login, or service name.	`-h` `--help`
Display the version number of the Server Control Utility. **Note**: This command does not require a machine name, login, or service name.	`-V` `--version`
Get information about the MicroStrategy network	
List machines that the Server Control Utility can see and affect. **Note**: This command does not require a machine name, login, or service name.	`lm` `list-machines`
List the MicroStrategy services available on a machine. **Note**: This command does not require a service name.	`ls` `list-servers`
List the ODBC DSNs available on a machine. **Note**: This command does not require a service name.	`lod` `list-odbc-dsn`

If you want to. . .	Then use this command. . .
Configure a service	
Display the configuration information for a service, in XML format. For more information, see *Using files to store output and provide input, page 24*. **Note**: You can optionally specify a file to save the configuration properties to.	`gsvc instancename [> filename.xml]` `get-service-configuration instancename [> filename.xml]`
Specify the configuration information for a service, in XML format. For more information, see *Using files to store output and provide input, page 24*. **Note**: You can optionally specify a file to read the configuration properties from.	`ssvc instancename [< filename.xml]` `set-service-configuration instancename [< filename.xml]`
Configure a server	
Display the configuration properties of a server, in XML format. For more information, see *Using files to store output and provide input, page 24*. **Note**: You can optionally specify a file to save the configuration properties to.	`gsc [> filename.xml]` `get-server-configuration [> filename.xml]`
Specify the configuration properties of a server, in XML format. For more information, see *Using files to store output and provide input, page 24*. **Note**: You can optionally specify a file to read the configuration properties from.	`ssc [< filename.xml]` `set-server-configuration [< filename.xml]`
Configure a server instance	
Display the configuration information for a server instance, in XML format. For more information, see *Using files to store output and provide input, page 24*. **Note**: You can optionally specify a file to save the configuration properties to.	`gsic instancename [> filename.xml]` `get-server-instance-configuration instancename [> filename.xml]`
Specify the configuration information for a server instance, in XML format. For more information, see *Using files to store output and provide input, page 24*. **Note**: You can optionally specify a file to read the configuration properties from.	`ssic instancename` `set-server-instance-configuration instancename [< filename.xml]`
Manage server instances	
Display the default instance for a service.	`gdi` `get-default-instance`
Set an instance of a service as the default instance.	`sdi instancename`

 © 2016, MicroStrategy Inc.

If you want to. . .	Then use this command. . .
	`set-default-instance` *`instancename`*
Create a new server instance.	`ci` *`instancename`* `create-instance` *`instancename`*
Create a copy of a server instance. Specify the name for the new instance as *`newinstancename`*.	`cpi` *`instancename newinstancename`* `copy-instance` *`instancename newinstancename`*
Delete a server instance.	`di` *`instancename`* `delete-instance` *`instancename`*
Register a server instance as a service.	`rs` *`instancename`* `register-service` *`instancename`*
Unregister a registered server instance as a service.	`us` *`instancename`* `unregister-service` *`instancename`*
Display the license information for a service instance.	`gl` *`instancename`* `get-license` *`instancename`*
Display the status information for a server instance	`gs` *`instancename`* `get-status` *`instancename`*
Start or stop a server instance	
Start a server instance as a service.	`start --service` *`instancename`*
Start a server instance as an application. For more information, see *Starting Intelligence Server as an application, page 19*.	`start --interactive` *`instancename`*
Stop a server instance that has been started as a service.	`stop` *`instancename`*
Pause a server instance that has been started as a. service	`pause` *`instancename`*
Resume a server instance that has been started as a service and paused.	`resume` *`instancename`*
Terminate a server instance that has been started as a service.	`term` *`instancename`* `terminate` *`instancename`*

Using files to store output and provide input

Certain Server Control Utility commands involve XML definitions. The commands to display a server configuration, a service configuration, and a server instance configuration all output an XML definition. The commands to modify a server configuration, a service configuration, and a server instance configuration all require an XML definition as input.

It is difficult and time consuming to type a complete server, service, or server instance configuration from the command line. An easier way to configure them is to output the current configuration to a file, modify the file with a text editor, and then use the file as input to a command to modify the configuration.

Configuring Intelligence Server with XML files requires extensive knowledge of the various parameters and values used to define Intelligence Server configurations. Providing an incorrect XML definition to configure Intelligence Server can cause errors and unexpected functionality.

For example, the following command saves the default server instance configuration to an XML file:

```
mstrctl -s IntelligenceServer
gsic > filename.xml
```

The server instance configuration is saved in the file `filename.xml`, in the current directory.

The following command modifies the default server instance configuration by reading input from an XML file:

```
mstrctl -s IntelligenceServer
ssic < filename.xml
```

The XML definition in `ServerInstance.xml` is used to define the server instance configuration.

Managing and monitoring projects

The System Administration Monitor lists all the projects on an Intelligence Server and all the machines in the cluster that Intelligence Server is using. You can monitor the status of the projects on a project source, and load, unload, idle, and resume projects for the entire project source or for a single node of the cluster. You can also schedule various system maintenance tasks from the Scheduled Maintenance view.

The System Administration group contains the following views:

- Project, which helps you keep track of the status of all the projects contained in the selected project source. For detailed information, see *Managing project status, configuration, or security: Project view, page 25*.

 © 2016, MicroStrategy Inc.

- Cluster, which helps you manage how projects are distributed across the servers in a cluster. For detailed information, see *Managing clustered Intelligence Servers: Cluster view, page 26*.
- The Scheduled Maintenance monitor, which lists all the scheduled maintenance tasks. For detailed information, see *Scheduling administrative tasks, page 485*.

Managing project status, configuration, or security: Project view

The Project view helps you keep track of the status of all the projects contained in the selected project source. It also enables access to a number of project maintenance interfaces in one place. This makes it faster and easier to perform maintenance tasks such as purging caches, managing security filters, or loading or unloading projects from Intelligence Server.

To access the Project view

1. Expand **Administration** in the project source's folder list.
2. Expand the **System Administration** group, and then select **Project**. The projects and their statuses display on the right-hand side.

Using the Project view

The Project view lists all the projects in the project source. If your system is set up as a cluster of servers, the Project Monitor displays all projects in the cluster, including the projects that are not running on the node from which you are accessing the Project Monitor. For details on projects in a clustered environment, see *Distributing projects across nodes in a cluster, page 388*.

To view the status of a project, select the **List** or **Details** view, and click the **+** sign next to the project's name. A list of all the servers in the cluster expands below the project's name. The status of the project on each server is shown next to the server's name. If your system is not clustered, there is only one server in this list.

For projects distributed asymmetrically across nodes of a cluster, a primary server is assigned to each project. A project's primary server handles the time-based scheduling for that project. The primary server is displayed in bold, and **Primary Server** appears after the server name.

From the Project view, you can access a number of administrative and maintenance functions. You can:

- Manage the users and security filters for a project
- View the change journal for a project (for details, see *Monitoring system activity: Change journaling, page 281*)

- Export and print the project's schema or other project documentation
- Load or unload projects from Intelligence Server, or idle or resume projects for maintenance (for details, see *Changing the status of a project, page 30*)

To load a project on a specific server in a cluster, you use the Cluster Monitor. For details on this procedure, see *Managing clustered Intelligence Servers: Cluster view, page 26*.

- Purge report, element, or object caches for projects

These tasks are all available by right-clicking a project in the Project Monitor. For more detailed information about any of these options, see the Help or related sections in this guide.

You can perform an action on multiple projects at the same time. To do this, select several projects (**CTRL**+click), then right-click and select one of the options.

You can also schedule any of these maintenance functions from the Schedule Administration Tasks dialog box. To access this dialog box, right-click a project in the Project view and select **Schedule Administration Tasks**. For more information, including detailed instructions on scheduling a task, see *Scheduling administrative tasks, page 485*.

Managing clustered Intelligence Servers: Cluster view

The Cluster view helps you keep track of the status of your clustered Intelligence Servers. Through the Cluster view, you can view the status of each node, add or remove nodes in the cluster, and view how projects are distributed across the nodes.

To access the Cluster view

1. Expand **Administration** in the project source's folder list.
2. Expand the **System Administration** group, and then select **Cluster**. The projects and their statuses display on the right-hand side.
3. To see a list of all the projects on a node, click the **+** sign next to that node. The status of the project on the selected server is shown next to the project's name.

Using the Cluster view

From the Cluster view, you can access a number of administrative and maintenance functions. You can:

- Manage the security policy settings for the project source
- Join or leave a cluster

 © 2016, MicroStrategy Inc.

- Manage the change journaling for projects on a cluster
- Purge the object cache for a server

These tasks are all available by right-clicking a server in the Cluster view.

You can also load or unload projects from a machine, or idle or resume projects on a machine for maintenance (for details, see *Changing the status of a project, page 30*) by right-clicking a project on a server. For more detailed information about any of these options, see the *MicroStrategy Developer Help*, or see *Managing your projects across nodes of a cluster, page 390*.

Setting the status of a project

Each project in Intelligence Server can operate in one of several modes. Project modes allow for various system administration tasks to occur without interrupting Intelligence Server operation for other projects. The tasks that are allowed to occur depend on the job or jobs that are required for that task.

A project's status can be one of the following:

- *Loaded, page 27*
- *Unloaded, page 27*
- *Request Idle, page 28*
- *Execution Idle, page 28*
- *Warehouse Execution Idle, page 29*
- *Full Idle, page 29*
- *Partial Idle, page 29*

For instructions on changing a project's status, see *Changing the status of a project, page 30*.

For example scenarios where the different project idle modes can help to support project and data warehouse maintenance tasks, see *Project and data warehouse maintenance example scenarios, page 31*.

Loaded

A project in Loaded mode appears as an available project in Developer and MicroStrategy Web products. In this mode, user requests are accepted and processed as normal.

Unloaded

Unloaded projects are still registered on Intelligence Server, but they do not appear as available projects in Developer or MicroStrategy Web products, even for administrators. Nothing can be done in the project until it is loaded again.

© 2016, MicroStrategy Inc.

Unloading a project can be helpful when an administrator has changed some project configuration settings that do not affect run-time execution and are to be applied to the project at a later time. The administrator can unload the project, and then reload the project when it is time to apply the project configuration settings.

A project unload request is fully processed only when all executing jobs for the project are complete.

Request Idle

Request Idle mode helps to achieve a graceful shutdown of the project rather than modifying a project from Loaded mode directly to Full Idle mode. In this mode, Intelligence Server:

- Stops accepting new user requests from the clients for the project.
- Completes jobs that are already being processed. If a user requested that results be sent to her History List, the results are available in her History List after the project is resumed.

Setting a project to Request Idle can be helpful to manage server load for projects on different clusters. For example, in a cluster with two nodes named Node1 and Node2, the administrator wants to redirect load temporarily to the project on Node2. The administrator must first set the project on Node1 to Request Idle. This allows existing requests to finish execution for the project on Node1, and then all new load is handled by the project on Node2.

Execution Idle

A project in Execution Idle mode is ideal for Intelligence Server maintenance because this mode restricts users in the project from running any job in Intelligence Server. In this mode, Intelligence Server:

- Stops executing all new and currently executing jobs and, in most cases, places them in the job queue. This includes jobs that require SQL to be submitted to the data warehouse and jobs that are executed in Intelligence Server, such as answering prompts.

  If a project is idled while Intelligence Server is in the process of fetching query results from the data warehouse for a job, that job is canceled instead of being placed in the job queue. When the project is resumed, if the job was sent to the user's History List, an error message is placed in the History List. The user can click the message to resubmit the job request.

- Allows users to continue to request jobs, but execution is not allowed and the jobs are placed in the job queue. Jobs in the job queue are displayed as "Waiting for project" in the Job Monitor. When the project is resumed, Intelligence Server resumes executing the jobs in the job queue.

This mode allows you to perform maintenance tasks for the project. For example, you can still view the different project administration monitors, create reports, create

 © 2016, MicroStrategy Inc.

attributes, and so on. However, tasks such as element browsing, exporting, and running reports that are not cached are not allowed.

Warehouse Execution Idle

A project in Warehouse Execution Idle mode is ideal for data warehouse maintenance because this mode restricts users in the project from running any SQL against the data warehouse. In this mode, Intelligence Server:

- Accepts new user requests from clients for the project, but it does not submit any SQL to the data warehouse.
- Stops any new or currently executing jobs that require SQL to be executed against the data warehouse and, in most cases, places them in the job queue. These jobs display as "Waiting for project" in the Job Monitor. When the project is resumed, Intelligence Server resumes executing the jobs in the job queue.

  If a project is idled while Intelligence Server is in the process of fetching query results from the data warehouse for a job, that job is canceled instead of being placed in the job queue. When the project is resumed, if the job was sent to the user's History List, an error message is placed in the History List. The user can click the message to resubmit the job request.
- Completes any jobs that do not require SQL to be executed against the data warehouse.

This mode allows you to perform maintenance tasks on the data warehouse while users continue to access non-database-dependent functionality. For example, users can run cached reports, but they cannot drill if that drilling requires additional SQL to be submitted to the data warehouse. Users can also export reports and documents in the project.

Full Idle

Full Idle is a combination of Request Idle and Execution Idle. In this mode, Intelligence Server does not accept any new user requests and active requests are canceled. When the project is resumed, Intelligence Server does not resubmit the canceled jobs and it places an error message in the user's History List. The user can click the message to resubmit the request.

This mode allows you to stop all Intelligence Server and data warehouse processing for a project. However, the project still remains in Intelligence Server memory.

Partial Idle

Partial Idle is a combination of Request Idle and Warehouse Execution Idle. In this mode, Intelligence Server does not accept any new user requests. Any active requests that require SQL to be submitted to the data warehouse are queued until the project is resumed. All other active requests are completed.

© 2016, MicroStrategy Inc.

This mode allows you to stop all Intelligence Server and data warehouse processing for a project, while not canceling jobs that do not require any warehouse processing. The project still remains in Intelligence Server memory.

Changing the status of a project

To load or unload a project

If the project is running on multiple clustered Intelligence Servers, the project is loaded or unloaded from all nodes. To load or unload the project from specific nodes, use the **Cluster** view instead of the **Project** view. For detailed instructions, see *Using the Cluster view, page 26*.

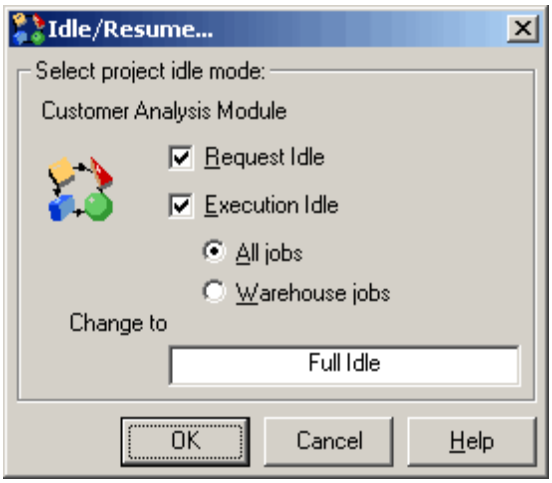
1. In Developer, log in to the project source containing the project.
2. Under that project source, expand **Administration**, then expand **System Administration**, and select **Project**.
3. Right-click the project, point to **Administer Project**, and select **Load** or **Unload**. The project is loaded or unloaded. If you are using clustered Intelligence Servers, the project is loaded or unloaded for all nodes in the cluster.

To idle or resume a project

If the project is running on multiple clustered Intelligence Servers, the project status changes for all nodes. To idle or resume the project on specific nodes, use the **Cluster** view instead of the **Project** view. For detailed instructions, see *Using the Cluster view, page 26*.

1. In Developer, log in to the project source containing the project.
2. Under that project source, expand **Administration**, then expand **System Administration**, and then select **Project**.
3. Right-click the project, point to Administer Project, and select **Idle/Resume**. The Idle/Resume dialog box opens.

 © 2016, MicroStrategy Inc.



4 Select the options for the idle mode that you want to set the project to:

- Request Idle (**Request Idle**): all executing and queued jobs finish executing, and any newly submitted jobs are rejected.
- Execution Idle (**Execution Idle** for **All Jobs**): all executing, queued, and newly submitted jobs are placed in the queue, to be executed when the project resumes.
- Warehouse Execution Idle (**Execution Idle** for **Warehouse jobs**): all executing, queued, and newly submitted jobs that require SQL to be submitted to the data warehouse are placed in the queue, to be executed when the project resumes. Any jobs that do not require SQL to be executed against the data warehouse are executed.
- Full Idle (**Request Idle** and **Execution Idle** for **All jobs**): all executing and queued jobs are canceled, and any newly submitted jobs are rejected.
- Partial Idle (**Request Idle** and **Execution Idle** for **Warehouse jobs**): all executing and queued jobs that do not submit SQL against the data warehouse are canceled, and any newly submitted jobs are rejected. Any currently executing and queued jobs that do not require SQL to be executed against the data warehouse are executed.

To resume the project from a previously idled state, clear the **Request Idle** and **Execution Idle** check boxes.

5 Click **OK**. The Idle/Resume dialog box closes and the project goes into the selected mode. If you are using clustered Intelligence Servers, the project mode is changed for all nodes in the cluster.

Project and data warehouse maintenance example scenarios

In addition to the example scenarios provided with the different project idle modes, the list below describes some other maintenance scenarios that can be achieved using

various project idle modes:

- Database maintenance for a data warehouse is scheduled to run at midnight, during which time the data warehouse must not be accessible to users. At 11:00 P.M., the administrator sets the project mode to Request Idle. All currently executing jobs will finish normally. At 11:30 P.M., the administrator sets the project mode to Warehouse Execution Idle, disallowing any execution against the data warehouse while maintenance tasks are performed. After maintenance is complete, the administrator sets the project to Loaded to allow normal execution and functionality to resume for the project.
- Two projects, named Project1 and Project 2, use the same data warehouse. Project1 needs dedicated access to the data warehouse for a specific length of time. The administrator first sets Project2 to Request Idle. After existing activity against the data warehouse is complete, Project2 is restricted against executing on the data warehouse. Then, the administrator sets Project2 to Warehouse Execution Idle mode to allow data warehouse-independent activity to execute. Project1 now has dedicated access to the data warehouse until Project2 is reset to Loaded.
- When the administrator schedules a project maintenance activity, the impact on users of the project during this time can be reduced. The administrator can set a project's idle mode to Request Idle, followed by Partial Idle, and finally to Full Idle. This process can reduce user access to a project and data warehouse gradually, rather than changing directly to Full Idle and thus immediately stopping all user activity.

Processing jobs

Any request submitted to Intelligence Server from any part of the MicroStrategy system is known as a job. Jobs may originate from servers such as Narrowcast Server or Intelligence Server's internal scheduler, or from client applications such as Developer, MicroStrategy Web, Mobile, Integrity Manager, or another custom-coded application.

The main types of requests include report execution requests, object browsing requests, element browsing requests, Report Services document requests, and HTML document requests.

The Job Monitor shows you which jobs are currently executing and lets you cancel jobs as necessary. For information about the job monitor, see *Monitoring currently executing jobs, page 47*.

By default, jobs are processed on a first-in first-out basis. However, your system probably has some jobs that need to be processed before other jobs. You can assign a priority level to each job according to factors such as the type of request, the user or user group requesting the job, the source of the job (such as Developer, Mobile, or MicroStrategy Web), the resource cost of the job, or the project containing the job. Jobs with a higher priority have precedence over jobs with a lower priority, and they are processed first if there is a limit on the resources available. For detailed information on job priority, including instructions on how to prioritize jobs, see *Prioritizing jobs, page 340*.

 © 2016, MicroStrategy Inc.

Intelligence Server job processing (common to all jobs)

Regardless of the type of request, Intelligence Server uses some common functionality to satisfy them. The following is a high-level overview of the processing that takes place.

1. A user makes a request from a client application such as MicroStrategy Web, which sends the request to Intelligence Server.
2. Intelligence Server determines what type of request it is and performs a variety of functions to prepare for processing.

   Depending on the request type, a task list is composed that determines what tasks must be accomplished to complete the job, that is, what components the job has to use within the server that handle things like asking the user to respond to a prompt, retrieving information from the metadata repository, executing SQL against a database, and so on. Each type of request has a different set of tasks in the task list.
3. The components in Intelligence Server perform different tasks in the task list, such as querying the data warehouse, until a final result is achieved.

   Those components are the stops the job makes in what is called a *pipeline*, a path that the job takes as Intelligence Server works on it.
4. The result is sent back to the client application, which presents the result to the user.

Most of the actual processing that takes place is done in steps 2 and 3 internally in Intelligence Server. Although the user request must be received and the final results must be delivered (steps 1 and 4), those are relatively simple tasks. It is more useful to explain how Intelligence Server works. Therefore, the rest of this section discusses Intelligence Server activity as it processes jobs. This includes:

- *Processing report execution, page 33*
- *Processing object browsing, page 37*
- *Processing element browsing, page 38*
- *Processing Report Services document execution, page 41*
- *Processing HTML document execution, page 42*
- *Client-specific job processing, page 44*

Being familiar with this material should help you to understand and interpret statistics, Enterprise Manager reports, and other log files available in the system. This may help you to know where to look for bottlenecks in the system and how you can tune the system to minimize their effects.

Processing report execution

Reports are perhaps the most common requests made of Intelligence Server. All report requests have the following pieces:

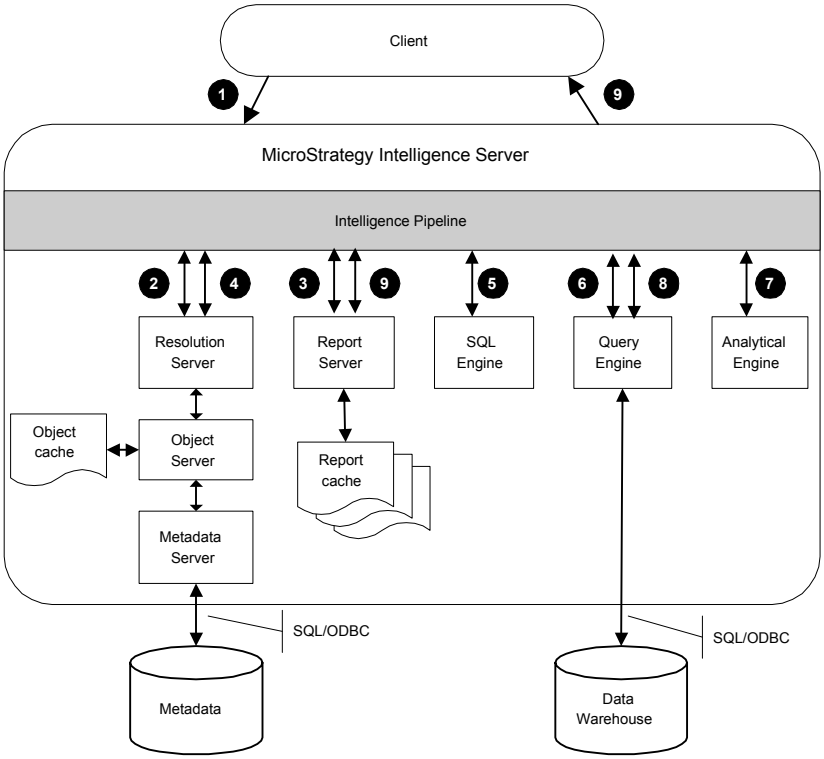
- A report instance is a container for all objects and information needed and produced during report execution including templates, filters, prompt answers, generated SQL, report results, and so on.
- A task list is a list of tasks that must be accomplished to complete a job. All jobs have a task list associated with them. Intelligence Server coordinates the report instance being passed from one internal Intelligence Server component to another as a report is executed.

The most prominent Intelligence Server components related to report job processing are listed here.

Component	Function
Analytical Engine Server	Performs complex calculations on a result set returned from the data warehouse, such as statistical and financial functions. Also, sorts raw results returned from the Query Engine into a cross-tabbed grid suitable for display to the user. In addition, it performs subtotal calculations on the result set. Depending on the metric definitions, the Analytical Engine will also perform metric calculations that were not or could not be performed using SQL, such as complex functions.
Metadata Server	Controls all access to the metadata for the entire project.
Object Server	Creates, modifies, saves, loads and deletes objects from metadata. Also maintains a server cache of recently used objects. The Object Server does not manipulate metadata directly. The Metadata Server does all reading/writing from/to the metadata; the Object Server uses the Metadata Server to make any changes to the metadata.
Query Engine	Sends the SQL generated by the SQL Engine to the data warehouse for execution.
Report Server	Creates and manages all server reporting instance objects. Maintains a cache of executed reports.
Resolution Server	Resolves prompts for report requests. Works in conjunction with Object Server and Element Server to retrieve necessary objects and elements for a given request.
SQL Engine Server	Generates the SQL needed for the report.

Below is a typical scenario of a report's execution within Intelligence Server. The diagram shows the report processing steps. An explanation of each step follows the diagram.

 © 2016, MicroStrategy Inc.



1. Intelligence Server receives the request.
2. The Resolution Server checks for prompts. If the report has one or more prompts, the user must answer them. For information about these extra steps, see *Processing reports with prompts, page 36*.
3. The Report Server checks the internal cache, if the caching feature is turned on, to see whether the report results already exist. If the report exists in the cache, Intelligence Server skips directly to the last step and delivers the report to the client. If no valid cache exists for the report, Intelligence Server creates the task list necessary to execute the report. For more information on caching, see *Result caches, page 400*.

   Prompts are resolved before the Server checks for caches. Users may be able to retrieve results from cache even if they have personalized the report with their own prompt answers.

4. The Resolution Server obtains the report definition and any other required application objects from the Object Server. The Object Server retrieves these objects from the object cache, if possible, or reads them from the metadata via the Metadata Server. Objects retrieved from metadata are stored in the object cache.
5. The SQL Generation Engine creates the optimized SQL specific to the RDBMS being used in the data warehouse. The SQL is generated according to the definition of the report and associated application objects retrieved in the previous step.
6. The Query Engine runs the SQL against the data warehouse. The report results are returned to Intelligence Server.

7 The Analytical Engine performs additional calculations as necessary. For most reports, this includes cross-tabbing the raw data and calculating subtotals. Some reports may require additional calculations that cannot be performed in the database via SQL.

8 Depending on the analytical complexity of the report, the results might be passed back to the Query Engine for further processing by the database until the final report is ready (in this case, steps 5–7 are repeated).

9 Intelligence Server's Report Server saves or updates the report in the cache, if the caching feature is turned on, and passes the formatted report back to the client, which displays the results to the user.

Processing reports with prompts

If the report has prompts, these steps are inserted in the regular report execution steps presented above (see *Processing report execution, page 33*):

1 Intelligence Server sends the job to the Resolution Server component. The Resolution Server discovers that the report definition contains a prompt and tells Intelligence Server to prompt the user for the necessary information.

2 Intelligence Server puts the job in a sleep mode and tells the Result Sender component to send a message to the client application prompting the user for the information.

3 The user completes the prompt, and the client application sends the user's prompt selections back to Intelligence Server.

4 Intelligence Server performs the security and governing checks and updates the statistics. It then *wakes up* the sleeping job, adds the user's prompt reply to the job's report instance, and passes the job to the Resolution Server again.

5 This cycle repeats until all prompts in the report are resolved.

A sleeping job times out after a certain period or if the connection to the client is lost. If the prompt reply comes back after the job has timed out, the user sees an error message.

All regular report processing resumes from the point at which Intelligence Server checks for a report cache, if the caching feature is turned on.

Processing personal Intelligent Cube reports

Personal Intelligent Cube reports are initially processed the same as a regular report, and the report instance is held in Intelligence Server's memory. If the user manipulates the report and that manipulation does not cause the base report's SQL to change, the Analytical Engine component services the request and sends the results to the client. No additional processing from the data warehouse is required.

Reports can also connect to Intelligent Cubes that can be shared by multiple reports. These Intelligent Cubes also allow the Analytical Engine to perform additional analysis without requiring any processing on the data warehouse.

 © 2016, MicroStrategy Inc.

For information on personal Intelligent Cubes and Intelligent Cubes, see the *In-memory Analytics Guide*.

Processing graph reports

When processing graph reports, Intelligence Server performs the regular report processing (see *Processing report execution, page 33*). Depending on the connection, the following happens:

- In a three-tier connection, Intelligence Server sends the report to Developer, which creates the graph image.
- In a four-tier connection, Intelligence Server uses the graph generation component to create the graph image and sends it to the client.

Processing object browsing

The definitions for all objects displayed in the folder list, such as folders, metrics, attributes, and reports, are stored in the metadata. Whenever you expand or select a folder in Developer or MicroStrategy Web, Intelligence Server must retrieve the objects from the metadata before it can display them in the folder list and the object viewer.

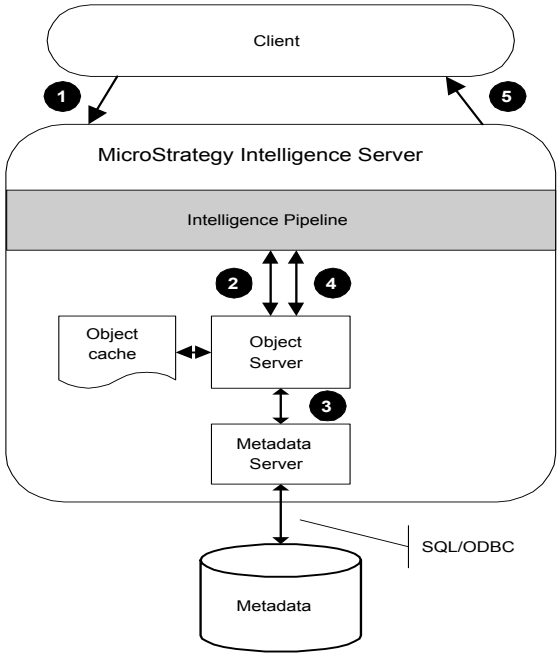
This process is called object browsing and it creates what are called object requests. It can cause a slight delay that you may notice the first time you expand or select a folder. The retrieved object definitions are then placed in Intelligence Server's memory (cache) so that the information is displayed immediately the next time you browse the same folder. This is called object caching. For more information on this, see *Object caches, page 451*.

The most prominent Intelligence Server components related to object browsing are listed here.

Component	Function
Metadata Server	Controls all access to the metadata for the entire project.
Object Server	Creates, modifies, saves, loads and deletes objects from metadata. Also maintains a server cache of recently used objects.
Source Net Server	Receives, de-serializes, and passes metadata object requests to the object server.

The diagram below shows the object request execution steps. An explanation of each step follows the diagram.

© 2016, MicroStrategy Inc.



1. Intelligence Server receives the request.
2. The Object Server checks for an object cache that can service the request. If an object cache exists, it is returned to the client and Intelligence Server skips to the last step in this process. If no object cache exists, the request is sent to the Metadata Server.
3. The Metadata Server reads the object definition from the metadata repository.
4. The requested objects are received by the Object Server where are they deposited into memory object cache.
5. Intelligence Server returns the objects to the client.

Processing element browsing

Attribute elements are typically stored in lookup tables in the data warehouse. This includes data that is unique to your business intelligence system, such as Northeast, Northwest, Central, and Asia in the Region attribute.

For a more thorough discussion of attribute elements, see the section in the Basic Reporting Guide about the logical data model.

When users request attribute elements from the system, they are said to be element browsing and create what are called element requests. More specifically, this happens when users:

- Answer prompts when executing a report
- Browse attribute elements in Developer using the Data Explorer (either in the Folder List or the Report Editor)

 © 2016, MicroStrategy Inc.

- Use Developer's Filter Editor, Custom Group Editor, or Security Filter Editor
- Use the Design Mode on MicroStrategy Web to edit the report filter

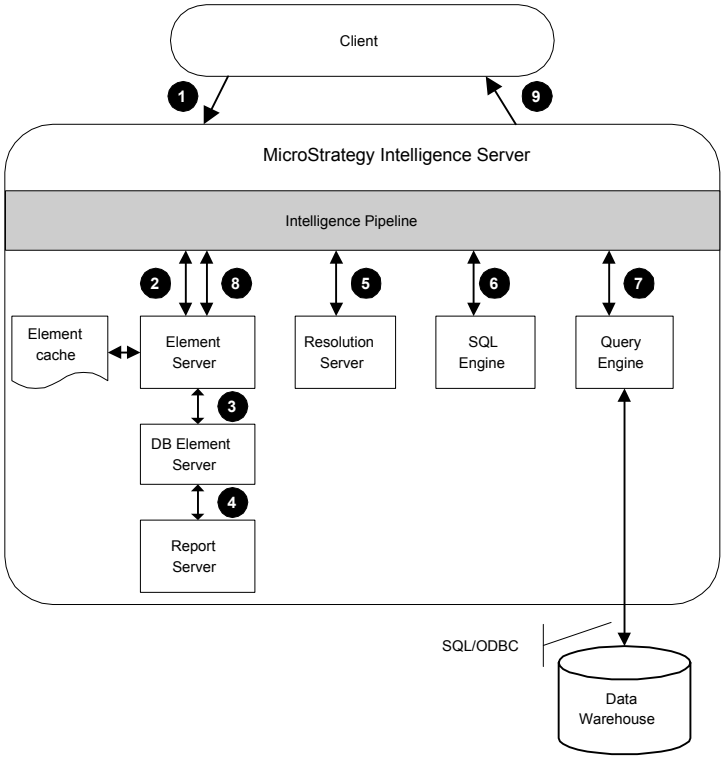
When Intelligence Server receives an element request from the user, it sends a SQL statement to the data warehouse requesting attribute elements. When it receives the results from the data warehouse, it then passes the results back to the user. Also, if the element caching feature is turned on, it stores the results in memory so that additional requests are retrieved from memory instead of querying the data warehouse again. For more information on this, see *Element caches, page 440*.

The most prominent Intelligence Server components related to element browsing are listed here.

Component	Function
DB Element Server	Transforms element requests into report requests and then sends report requests to the warehouse.
Element Net Server	Receives, de-serializes, and passes element request messages to the Element Server.
Element Server	Creates and stores server element caches in memory. Manages all element requests in the project.
Query Engine	Sends the SQL generated by the SQL Engine to the data warehouse for execution.
Report Server	Creates and manages all server reporting instance objects. Maintains a cache of executed reports.
Resolution Server	Resolves prompts for report requests. Works in conjunction with Object Server and Element Server to retrieve necessary objects and elements for a given request.
SQL Engine Server	Generates the SQL needed for the report.

The diagram below shows the element request execution steps. An explanation of each step follows the diagram.

© 2016, MicroStrategy Inc.



1. Intelligence Server receives the request.
2. The Element Server checks for a server element cache that can service the request. If a server element cache exists, the element cache is returned to the client. Skip to the last step in this process.
3. If no server element cache exists, the database Element Server receives the request and transforms it into a report request.

   The element request at this point is processed like a report request: Intelligence Server creates a report that has only the attributes and possibly some filtering criteria, and SQL is generated and executed like any other report.
4. The Report Server receives the request and creates a report instance.
5. The Resolution Server receives the request and determines what elements are needed to satisfy the request, and then passes the request to the SQL Engine Server.
6. The SQL Engine Server generates the necessary SQL to satisfy the request and passes it to the Query Engine Server.
7. The Query Engine Server sends the SQL to the data warehouse.
8. The elements are returned from the data warehouse to Intelligence Server and deposited in the server memory element cache by the Element Server.
9. Intelligence Server returns the elements to the client.

 © 2016, MicroStrategy Inc.

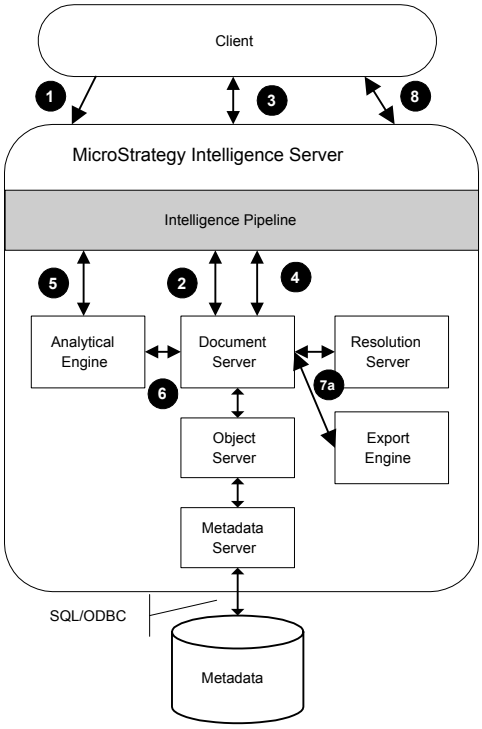
Processing Report Services document execution

A MicroStrategy Report Services document contains objects representing data coming from one or more reports. The document also holds positioning and formatting information. A document is used to combine data from multiple reports into a single display of presentation quality. When you create a document, you can specify the data that appears and can also control the layout, formatting, grouping, and subtotaling of that data. In addition, you can insert pictures into the document and draw borders on it. All these capabilities allow you to create documents that are suitable to present to management.

Most of the data on a document is from an underlying dataset. A dataset is a MicroStrategy report that defines the information that Intelligence Server retrieves from the data warehouse or cache. Other data that does not originate from the dataset is stored in the document's definition.

Document execution is slightly different from the execution of a single report, since documents can contain multiple reports.

The following diagram shows the document processing execution steps. An explanation of each step follows the diagram.



1 Intelligence Server receives a document execution request and creates a document instance in Intelligence Server. This instance holds the results of the request.

A document instance facilitates the processing of the document through Intelligence Server, similar to a report instance that is used to process reports. It contains the report instances for all the dataset reports and therefore has access to all the information that may be included in the dataset reports. This information includes prompts, formats, and so on.

2 The Document Server inspects all dataset reports and prepares for execution. It consolidates all prompts from from datasets into a single prompt to be answered. All identical prompts are merged so that the resulting prompt contains only one copy of each prompt question.

3 The Document Server, with the assistance of the Resolution Server, asks the user to answer the consolidated prompt. The user's answers are stored in the Document Server.

4 The Document Server creates an individual report execution job for each dataset report. Each job is processed by Intelligence Server, using the report execution flow described in *Processing report execution, page 33*. Prompt answers are provided by the Document Server to avoid further prompt resolution.

5 After Intelligence Server has completed all the report execution jobs, the Analytical Engine receives the corresponding report instances to begin the data preparation step. Document elements are mapped to the corresponding report instance to construct internal data views for each element.

Document elements include grouping, data fields, Grid/Graphs, and so on.

6 The Analytical Engine evaluates each data view and performs the calculations that are required to prepare a consolidated dataset for the entire document instance. These calculations include calculated expressions, derived metrics, and conditional formatting. The consolidated dataset determines the number of elements for each group and the number of detail sections.

7 The Document Server receives the final document instance to finalize the document format:

- Additional formatting steps are required if the document is exported to PDF or Excel format. The export generation takes place on the client side in three-tier and on the server side in four-tier, although the component in charge is the same in both cases.
- If the document is executed in HTML, the MicroStrategy Web client requests an XML representation of the document to process it and render the final output.

8 The completed document is returned to the client.

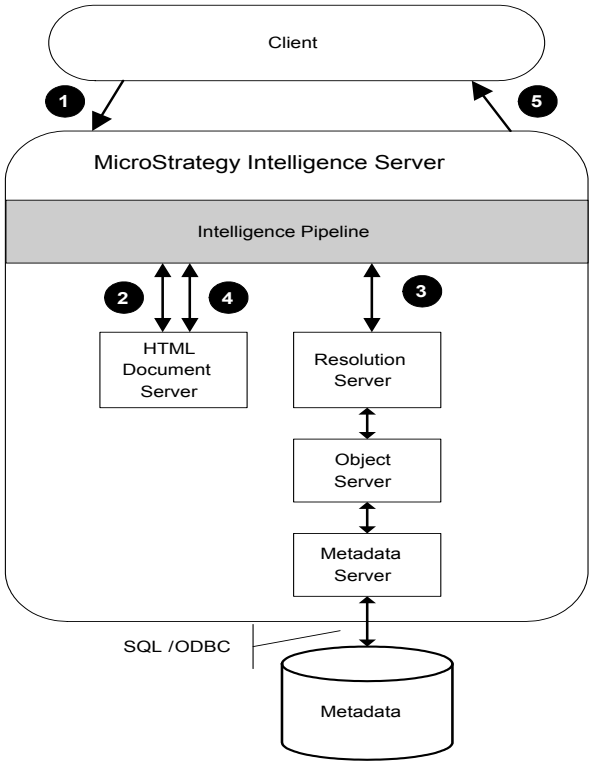
Processing HTML document execution

An HTML document is a container for formatting, displaying, and distributing multiple reports from a single request. HTML Documents are based on an HTML template, which allows them to contain any combination of text, images, hyperlinks, tables, grid reports, and graph reports. Any reports included in an HTML document are called the child reports of the HTML document.

Because HTML documents are collections of multiple reports, their execution process is slightly different from single reports. The most notable differences are shown in the procedure below.

 © 2016, MicroStrategy Inc.

The diagram below shows the HTML document processing execution steps. An explanation of each step follows the diagram.



1. Intelligence Server receives an HTML document execution request and creates an HTML document instance to go through Intelligence Server and hold the results.

   An HTML document instance facilitates the processing of the HTML document through Intelligence Server like a report instance is used for processing reports. It contains the report instances for all the child reports, the XML results for the child reports, and any prompt information that may be included in the child reports.

2. The HTML Document Server consolidates all prompts from child reports into a single prompt to be answered. Any identical prompts are merged so that the resulting single prompt contains only one copy of each prompt question.

3. Resolution Server asks the user to answer the consolidated prompt. (The user only needs to answer a single set of questions.)

4. The HTML Document Server splits the HTML document request into separate individual jobs for the constituent reports. Each report goes through the report execution flow as described above.

   Prompts have already been resolved for the child reports.

5. The completed request is returned to the client.

Client-specific job processing

This section explains the job processing steps that certain client applications perform as they deliver user requests to Intelligence Server. It also covers how those clients receive results, and how the results are displayed them to the user. These discussions cover the following clients:

- *Processing jobs from MicroStrategy Web products, page 44*
- *Processing jobs from Narrowcast Server, page 46*

For information about the processing steps performed by Intelligence Server for all jobs, see *Intelligence Server job processing (common to all jobs), page 33*.

Processing jobs from MicroStrategy Web products

This section provides a high-level overview of processing flow for requests originating in MicroStrategy Web or Web Universal. It also includes the job process for exporting reports in various formats.

Job requests from MicroStrategy Web products

1. The user makes a request from a web browser. The request is sent to the web server via HTTP or HTTPS.
2. An ASP.Net page or a servlet receives the request and calls the MicroStrategy Web API.
3. The MicroStrategy Web API sends the request to Intelligence Server, which processes the job as usual (see *Processing report execution, page 33*).
4. Intelligence Server sends the results back to the MicroStrategy Web API via XML.
5. MicroStrategy Web converts the XML to HTML within the application code:
   - In MicroStrategy Web, the conversion is primarily performed in ASP code.
   - In MicroStrategy Web Universal, the conversion is performed within the Java transform classes.
   - In some customizations, the conversion may occur within custom XSL classes. By default, the product does not use XSL for rendering output, except in document objects.
6. MicroStrategy Web sends the HTML to the client's browser, which displays the results.

What happens when I export a report from MicroStrategy Web?

Exporting a report from MicroStrategy Web products lets users save the report in another format that may provide additional capabilities for sharing, printing, or further manipulation. This section explains the additional processing the system must do when

 © 2016, MicroStrategy Inc.

exporting a report in one of several formats. This may help you to understand when certain parts of the MicroStrategy platform are stressed when exporting.

Reports in MicroStrategy Web can be exported to the following formats:

- *Export to Comma Separated File (CSV) or Excel with Plain Text, page 45*
- *Export to Excel with Formatting, page 45*
- *Export to PDF, page 46*

Exporting a report from MicroStrategy Web products causes Intelligence Server to retrieve the entire result set (no incremental fetch) into memory and send it to MicroStrategy Web. This increases the memory use on the Intelligence Server machine and it increases network traffic.

For information about governing report size limits for exporting, see *Limiting the information displayed at one time, page 348* and the following sections.

Export to Comma Separated File (CSV) or Excel with Plain Text

Export to Comma Separated File (CSV) and Export to Excel with Plain Text is done completely on Intelligence Server. These formats contain only report data and no formatting information. The only difference between these two formats is the internal "container" that is used.

The MicroStrategy system performs these steps when exporting to CSV or to Excel with plain text:

1. MicroStrategy Web product receives the request for the export and passes the request to Intelligence Server. Intelligence Server takes the XML containing the report data and parses it for separators, headers and metric values.
2. Intelligence Server then outputs the titles of the units in the Row axis. All these units end up in the same row of the result text.
3. Intelligence Server then outputs the title and header of one unit in the Column axis.
4. Step 3 is repeated until all units in the Column axis are completed.
5. Intelligence Server outputs all the headers of the Row axis and all metric values one row at a time.
6. The finished result is then passed to be output as a CSV or an Excel file, which is then passed to the client browser.

Export to Excel with Formatting

Exporting to Excel with formatting allows for reports to be exported to an Excel file and contain the same formatting as shown in the browser window. The report retains all cell coloring, font sizes, styles, and other formatting aspects.

Note the following:

- To export to Excel with formatting, the client machine must have Excel 2000 SR-1 or later.
- To export to Excel, users must first set their Export preferences by clicking **Preferences**, then **User preferences**, then **Export**, and select the Excel version they want to export to.

The MicroStrategy system performs these steps when exporting to Excel with formatting:

1. MicroStrategy Web product receives the request for the export to Excel and passes the request to Intelligence Server. Intelligence Server produces an HTML document by combining the XML containing the report data with the XSL containing formatting information.
2. Intelligence Server passes the HTML document to MicroStrategy Web, which creates an Excel file and sends it to the browser.
3. Users can then choose to view the Excel file or save it depending on the client machine operating system's setting for viewing Excel files.

Export to PDF

Exporting to PDF uses Intelligence Server's export engine to create a PDF (Portable Document Format) file. PDF files are viewed with Adobe's Acrobat reader and provide greater printing functionality than simply printing the report from the browser.

To view the PDF files, the client machine must have Adobe Acrobat Reader 5.0 version or greater.

Processing jobs from Narrowcast Server

MicroStrategy Narrowcast Server performs the following steps to deliver reports to users.

For detailed information about Narrowcast Server, see the *MicroStrategy Narrowcast Server Getting Started Guide*.

Job requests from MicroStrategy Narrowcast Server

1. A Narrowcast service execution is triggered by a schedule or external API call.
2. Narrowcast Server determines the service recipients and allocates work to Execution Engine (EE) machines.
3. EE machines determine personalized reports to be created for each recipient by using recipient preferences.
4. Narrowcast Server submits one report per user or one multipage report for multiple users, depending on service definition.

 © 2016, MicroStrategy Inc.

5 Intelligence Server processes the report job request as usual. (See *Processing report execution, page 33*.) It then sends the result back to Narrowcast Server.

6 Narrowcast Server creates formatted documents using the personalized report data.

7 Narrowcast Server packages documents as appropriate for the service's delivery method, such as e-mail, wireless, and so on.

8 Narrowcast Server delivers the information to recipients by the chosen delivery method.

Monitoring currently executing jobs

The Job Monitor informs you of what is happening with system tasks. However, it does not display detailed sub-steps that a job is performing. You can see jobs that are:

- Executing
- Waiting in the queue
- Waiting for a user to reply to a prompt
- Canceling
- Not completing because of an error

The Job Monitor displays which tasks are running on an Intelligence Server. When a job has completed it no longer appears in the monitor. You can view a job's identification number; the user who submitted it; the job's status; a description of the status and the name of the report, document, or query; and the project executing it.

To view the currently executing jobs

1 In Developer, log in to a project source. You must log in as a user with the Monitor Jobs privilege.

2 Expand **Administration**, then expand **System Monitors**, and then select **Jobs**. The job information displays on the right-hand side.

3 Because the Job Monitor does not refresh itself, you must periodically refresh it to see the latest status of jobs. To do this, press **F5**.

4 To view a job's details including its SQL, double-click it. A Quick View dialog box opens.

5 To view more details for all jobs displayed, right-click in the Job Monitor and select **View options**. Select the additional columns to display and click **OK**.

At times, you may see "Temp client" in the Network Address column. This may happen when Intelligence Server is under a heavy load and a user accesses the

list of available projects. Intelligence Server creates a temporary session that submits a job request for the available projects and then sends the list to the MicroStrategy Web client for display. This temporary session, which remains open until the request is fulfilled, is displayed as Temp client.

To cancel a job

1. Select the job in the Job Monitor.
2. Press **DELETE**, and then confirm whether you wish to cancel the job.

Using automated installation techniques

You can make installing the MicroStrategy system across your enterprise easier in several ways. They are mentioned here but more fully explained in theInstallation and Configuration Guide.

Using a Response file to install the product

The response file installation allows you to automate certain aspects of the installation by configuring a Windows INI-like response file, called `response.ini`. This option is typically implemented by Original Equipment Manufacturer (OEM) applications that embed MicroStrategy installations in other products. It can also be implemented by IT departments that want to have more control over desktop installations. For more information on how to set up and use a response file, see the Installation and Configuration Guide.

Using a Response file to configure the product

You can also use a response file to automate certain aspects of the MicroStrategy configuration. This response file supplies parameters to the Configuration Wizard to set up a metadata repository and statistics tables, Intelligence Server, and multiple project sources. For steps on setting up and using a response file for the Configuration Wizard, see the Installation and Configuration Guide.

Running a silent installation

Silent installations do not present any graphical user interface (GUI). They are typically implemented by IT departments that perform software distribution and installation across the network, for example, by using Microsoft's System Management Server software. This involves configuring a `setup.iss` file that the MicroStrategy Installation

 © 2016, MicroStrategy Inc.

Wizard uses. For steps on setting up and using a `setup.iss` file for a silent MicroStrategy installation, see the Installation and Configuration Guide.

OEMs may use silent installations; however, it is more common for OEMs to use a response file installation.

Security checklist before deploying the system

Use the checklist below to make sure you have implemented the appropriate security services or features for your system before it is deployed. All the security implementations listed below are described in detail in the preceding sections of this chapter.

Security implementation	Completed
Ensure that the Administrator password has been changed. When you install Intelligence Server, the Administrator account comes with a blank password that must be changed.	
Set up access controls for the database (see *Controlling access to data, page 73*). Depending on your security requirements you may need to:	
• Set up security views to restrict access to specific tables, rows, or columns in the database	
• Split tables in the database to control user access to data by separating a logical data set into multiple physical tables, which require separate permissions for access	
• Implement connection mapping to control individual access to the database	
• Configure passthrough execution to control individual access to the database from each project, and to track which users are accessing the RDBMS	
• Assign security filters to users or groups to control access to specific data (these operate similarly to security views but at the application level)	
Understand the MicroStrategy user model (see *The MicroStrategy user model, page 52*). Use this model to:	
• Select and implement a system authentication mode to identify users	
• Set up security roles for users and groups to assign basic privileges and permissions	
• Understand ACLs (access control lists), which allow users access permissions to individual objects	
• Check and, if necessary, modify privileges and permissions for anonymous authentication for guest users. (By default, anonymous access is disabled at both the server and the project levels.) Do not assign delete privileges to the guest user account.	
Assign privileges and permissions to control user access to application functionary. You may need to:	

© 2016, MicroStrategy Inc.

Security implementation	Completed
• Assign the Denied All permission to a special user or group so that, even if permission is granted at another level, permission is still denied	
• Make sure guest users (anonymous authentication) have access to the Log folder in C:\Program Files (x86)\Common Files\MicroStrategy. This ensures that any application errors that occur while a guest user is logged in can be written to the log files.	
Use your web application server security features to:	
• Implement file-level security requirements	
• Create security roles for the application server	
Make use of standard Internet security technologies such as firewalls, digital certificates, and encryption. For example:	
• Enable encryption for MicroStrategy Web products. By default most encryption technologies are not used unless you enable them.	
• If you are working with sensitive or confidential data, enable the setting to encrypt all communication between MicroStrategy Web server and Intelligence Server. **Note**: There may be a noticeable performance degradation because the system must encrypt and decrypt all network traffic.	
Locate the physical machine hosting the MicroStrategy Web application in a physically secure location.	
Restrict access to files stored on the machine hosting the MicroStrategy Web application by implementing standard file-level security offered by your operating system. Specifically, apply this type of security to protect access to the MicroStrategy administrator pages, to prevent someone from typing specific URLs into a browser to access these pages. (The default location of the Admin page file is `C:\Program Files (x86)\MicroStrategy\Web ASPx\asp\Admin.aspx`.) Be sure to restrict access to:	
• The asp directory	
• Admin.aspx	

 © 2016, MicroStrategy Inc.

2

SETTING UP USER SECURITY

Introduction

Security is a concern in any organization. The data warehouse may contain sensitive information that should not be viewed by all users. It is your responsibility as administrator to make the right data available to the right users.

MicroStrategy has a robust security model that enables you to create users and groups, and control what data they can see and what objects they can use. The security model is covered in the following sections:

- *The MicroStrategy user model, page 52*
- *Controlling access to application functionality, page 58*
- *Controlling access to data, page 73*
- *Merging users or groups, page 93*

Authentication, the process by which the system identifies the user, is an integral part of any security model. Authenticating users is addressed in *Chapter , Identifying Users: Authentication*.

© 2016, MicroStrategy Inc.

The MicroStrategy user model

This section provides an overview of what users and groups are in the system and how they can be imported or created.

About MicroStrategy users

Like most security architectures, the MicroStrategy security model is built around the concept of a user. To do anything useful with MicroStrategy, a user must log in to the system using a login ID and password. The user can then perform tasks such as creating objects or executing reports and documents, and can generally take advantage of all the other features of the MicroStrategy system.

MicroStrategy supports a single sign-on for users in an enterprise environment that consists of multiple applications, data sources, and systems. Users can log in to the system once and access all the resources of the enterprise seamlessly. For more details about implementing single sign-on in MicroStrategy, see *Enabling single sign-on authentication, page 132*.

Users are defined in the MicroStrategy metadata, and exist across projects. You do not have to define users for every project you create in a single metadata repository.

Each user has a unique profile folder in each project. This profile folder appears to the user as the “My Personal Objects” folder. By default other users’ profile folders are hidden. They can be viewed by, in the Developer Preferences dialog box, in the Developer: Browsing category, selecting the **Display Hidden Objects** check box.

Administrator is a built-in default user created with a new MicroStrategy metadata repository. The Administrator user has all privileges and permissions for all projects and all objects.

One of the first things you should do in your MicroStrategy installation is to change the password for the Administrator user.

About MicroStrategy user groups

A user group (or “group” for short) is a collection of users. Groups provide a convenient way to manage a large number of users.

Instead of assigning privileges, such as the ability to create reports, to hundreds of users individually, you may assign privileges to a group. Groups may also be assigned permissions to objects, such as the ability to add reports to a folder.

In addition to having privileges of their own, subgroups always inherit the privileges from their parent groups.

The following groups are provided with your MicroStrategy installation:

 © 2016, MicroStrategy Inc.

- *The Everyone group, page 53*
- *Authentication-related groups, page 53*
- *Groups corresponding to product offerings, page 54*
- *Administrator groups, page 54*

For a list of the privileges assigned to each group, see the *List of Privileges* chapter in the Supplemental Reference for System Administration.

Do not modify the privileges for an out-of-the-box user group. During upgrades to newer versions of MicroStrategy, the privileges for the out-of-the-box user groups are overwritten with the default privileges. Instead, you should copy the user group you need to modify and make changes to the copied version.

The Everyone group

All users except for guest users are automatically members of the Everyone group. The Everyone group is provided to make it easy for you to assign privileges, security role memberships, and permissions to all users.

When a project is upgraded from MicroStrategy version 7.5.x or earlier to version 9.x, the Use Developer privilege is automatically granted to the Everyone group. This ensures that all users who were able to access Developer in previous versions can continue to do so.

Authentication-related groups

These groups are provided to assist you in managing the different ways in which users can log into the MicroStrategy system. For details on the different authentication methods, see *Chapter , Identifying Users: Authentication*.

- **Public/Guest**: The Public group provides the capability for anonymous logins and is used to manage the access rights of guest users. If you choose to allow anonymous authentication, each guest user assumes the profile defined by the Public group. When a user logs in as a guest, a new user is created dynamically and becomes a member of the Public group. For more information about anonymous authentication and the Public/Guest group, see *Implementing anonymous authentication, page 102*.
- **3rd Party Users**: Users who access MicroStrategy projects through third-party (OEM) software.
- **LDAP Users**: The group into which users that are imported from an LDAP server are added.
- **LDAP Public/Guest**: The group that is used when a user is linked to an LDAP account but not imported.

For information on integrating LDAP with MicroStrategy, see *Implementing LDAP authentication, page 105*.

- **Warehouse Users**: Users who access a project through a warehouse connection.

Groups corresponding to product offerings

These groups are built-in groups that correspond to the licenses you have purchased. Using these groups gives you a convenient way to assign product-specific privileges.

- **Architect**: Architects function as project designers and can create attributes, facts, hierarchies, projects, and so on.
- **Analyst**: Analysts have the privileges to execute simple reports, answer prompts, drill on reports, format reports, create reports by manipulating Report Objects, create derived metrics, modify view filter, pivot reports, create page by, and sort using advanced options.
- **Developer**: Developers can design new reports from scratch, and create report components such as consolidations, custom groups, data marts, documents, drill maps, filters, metrics, prompts, and templates.
- **Web Reporter**: Web Reporters can view scheduled reports and interactively slice and dice them. They can also use the printing, exporting, and e-mail subscription features.
- **Web Analyst**: Web Analysts can create new reports with basic report functionality, and use ad hoc analysis from Intelligent Cubes with interactive, slice and dice OLAP.
- **Web Professional**: Web Professional users have the maximum access to MicroStrategy Web functionality. They can create Intelligent Cubes and reports for users, with full reporting, ad hoc, and OLAP capabilities with seamless ROLAP analysis.

Administrator groups

- **System Monitors**: The System Monitors groups provide an easy way to give users basic administrative privileges for all projects in the system. Users in the System Monitors groups have access to the various monitoring and administrative monitoring tools
    - **System Administrators**: The System Administrators group is a group within the System Monitors group. It provides all the capabilities of the System Monitors group plus the ability to modify configuration objects such as database instances, and so on.

Privileges

Privileges allow users to access and work with various functionality within the software. All users created in the MicroStrategy system are assigned a set of privileges by default.

 © 2016, MicroStrategy Inc.

For detailed information about privileges, including how to assign privileges to a user or group, see *Controlling access to functionality: Privileges, page 66*. For a list of all user and group privileges in MicroStrategy, see the *List of Privileges* chapter in the Supplemental Reference for System Administration.

To see which users are using certain privileges, use the License Manager. See *Using License Manager, page 207*.

To view a user's privileges

1. In Developer, log into a project source. You must log in as a user with the Create And Edit Users And Groups privilege.
2. Expand **Administration**, then **User Manager**, and then the group containing the user. A list of users in the selected group appears in the main pane of Developer.
3. Right-click the user and select **Grant access to projects**. The User Editor opens to the Project Access dialog box. The privileges that the user has for each project are listed, as well as the source of those privileges (inherent to user, inherited from a group, or inherited from a security role).

Permissions

Permissions allow users to interact with various objects in the MicroStrategy system. All users created in the MicroStrategy system have certain access rights to certain objects by default.

Permissions differ from privileges in that permissions restrict or allow actions related to a single object, while privileges restrict or allow actions across all objects in a project.

For detailed information about permissions, including how to assign permissions for an object to a user or group, see *Controlling access to objects: Permissions, page 58*.

To view the permissions for an object

1. From within Developer, right-click the object and select **Properties**. The Properties dialog box opens.
2. Expand the **Security** category. The dialog box lists all users and groups with access to the object, and what permissions those users and groups have for the object.

© 2016, MicroStrategy Inc.

Creating, importing, and deleting users and groups

It is possible to create users individually using the User Manager interface in Developer, or using Command Manager (for a detailed explanation of how to use Command Manager, including examples, see *Chapter 15, Automating Administrative Tasks with Command Manager*). You can also import users and groups from a text file, from a Windows user directory, or from an LDAP directory.

To create a new user with the User Editor in Developer

1. In Developer, log into a project source. You must log in as a user with the Create And Edit Users And Groups privilege.
2. Expand **Administration**, then **User Manager**, and then a group that you want the new user to be a member of. If you do not want the user to be a member of a group, select **Everyone**.
3. From the **File** menu, point to **New** and then select **User**. The User Editor opens.
4. Specify the user information for each category in the editor. For details about each field, see the *MicroStrategy Developer Help*.

The user login ID is limited to 50 characters.

For detailed information about other methods for creating or importing users or groups, see the *MicroStrategy Developer Help*.

To delete a user

If a Narrowcast user exists that inherits authentication from the user that you are deleting, you must also remove the authentication definition from that Narrowcast user. For instructions, see the *MicroStrategy Narrowcast Server Administration Guide*.

1. In Developer, log into a project source. You must log in as a user with the Create And Edit Users And Groups privilege.
2. Expand **Administration**, then **User Manager**, and then browse to the group containing the user.
3. Select the user and press **DELETE**.
4. A dialog box opens, asking you to confirm the action. Click **OK** to continue.
5. If the user owns a profile folder, a dialog box opens asking if you want to delete the user's profile folder:

 © 2016, MicroStrategy Inc.

- If you click **No**, the folder and its contents remain on the system and ownership is assigned to Administrator. You may later assign ownership and access control lists for the folder and its contents to other users.
- If you click **Yes**, the folder and all of its contents are deleted.

Monitoring users' connections to projects

When a user connects to a project, a user connection is established. You may want to see a list of all users connected to projects within a project source. The User Connection Monitor displays a list of all connections and allows you to disconnect a user.

To view the active user connections

1. In Developer, log in to a project source. You must log in as a user with the Monitor User Connections privilege.
2. Expand **Administration**, then expand **System Monitors**, and then select **User Connections**. The user connection information displays on the right-hand side. For each user, there is one connection for each project the user is logged in to, plus one connection for `<Server>` indicating that the user is logged in to the project source.

Note the following:

— **Scheduler**: Connections made by Intelligence Server to process scheduled reports or documents appear as `<Scheduler>` in the Network Address column. Scheduler sessions cannot be manually disconnected as described above. However, these sessions will be removed automatically by Intelligence Server when the user session idle time out value is reached.

— **Temp client**: At times, you may see "Temp client" in the Network Address column. This may happen when Intelligence Server is under a heavy load and a user accesses the Projects or Home page in MicroStrategy Web (the pages that display the list of available projects). Intelligence Server creates a temporary session that submits a job request for the available projects and then sends the list to the MicroStrategy Web client for display. This temporary session, which remains open until the request is fulfilled, is displayed as "Temp client."

3. To view a connection's details, double-click it. A Quick View dialog box opens.

To disconnect a user

If you disconnect users from the project source (the <Configuration> entry in the User Connection Monitor), they are also disconnected from any projects

they were connected to.

1. In the User Connection Monitor, select the connection.
2. Press **DELETE**. Confirm that you wish to disconnect the user.

Controlling access to application functionality

Access control governs the resources that an authenticated user can read, modify, or write. In addition to controlling access to data (see *Controlling access to data, page 73*), you must also control access to application functionality, such as the ability to create reports or which reports are viewable. The MicroStrategy system provides a rich set of functionality for access control within Intelligence Server:

- *Controlling access to objects: Permissions, page 58*
- *Controlling access to functionality: Privileges, page 66*
- *Defining sets of privileges: Security roles, page 69*

Controlling access to objects: Permissions

Permissions define the degree of control users have over individual objects in the system. For example, in the case of a report, a user may have permission to view the report definition and execute the report, but not to modify the report definition or delete the report.

While privileges are assigned to users (either individually, through groups, or with security roles), permissions are assigned to objects. More precisely, each object has an Access Control List (ACL) that specifies which permissions different sets of users have on that object.

Intelligence Server includes special privileges called Bypass All Object Security Access Checks and Bypass Schema Object Security Access Checks. Users with these privileges are not restricted by access control permissions and are considered to have full control over all objects and schema objects, respectively. For information about privileges, see *Controlling access to functionality: Privileges, page 66*.

To modify permissions for an object in Developer

1. In Developer, right-click the object and select **Properties**. The Properties dialog box for that object opens.

 © 2016, MicroStrategy Inc.

To modify an object's ACL, you must access the Properties dialog box directly from Developer. If you access the Properties dialog box from within an editor, you can view the object's ACL but cannot make any changes.

2 Select the **Security** category.

3 For the User or Group (click **Add** to select a new user or group), from the **Object** drop-down list, select the predefined set of permissions, or select **Custom** to define a custom set of permissions. If the object is a folder, you can also assign permissions to objects contained in that folder using the **Children** drop-down list.

4 Click **OK**.

For specific information about each setting in the dialog box, press F1 to see the Help.

To modify permissions for an object in MicroStrategy Web

1 In MicroStrategy Web, right-click an object and select **Share**. The Share dialog box for that object opens.

2 To modify permissions for a user or group, from the **Permission Level** drop-down list for that user or group, select the predefined set of permissions, or select **Custom** to define a custom set of permissions.

3 To add new users or groups to the object's access control list (ACL):

   a Click **Choose Users/Groups**. The User/Group browser opens.

   b Select the users or groups that you want to add to the object's ACL.

   c From the **Choose a Permission Level** drop-down list, select the predefined set of permissions, or select **Custom** to define a custom set of permissions.

   d Click **Add**. The users or groups are added to the list of users or groups, with the specified permissions.

4 To remove a user or group from the object's ACL, click the **X** next to the user or group's name.

5 When you are finished modifying the object's permissions, click **OK**. The dialog box closes and your changes are saved.

Access control list (ACL)

The Access Control List (ACL) of an object is a list of users and groups, and the access permissions that each has for the object.

For example, for the Northeast Region Sales report you can specify the following permissions:

- The Managers and Executive user groups have View access to the report.

- The Developers user group (people who create and modify your applications) has Modify access.
- The Administrators user group has Full Control of the report.
- The Everyone user group (any user not in one of the other groups) should have no access to the report at all, so you assign the Denied All permission grouping.

The default ACL of a newly created object has the following characteristics:

- The owner (the user who created the object) has Full Control permission.
- Permissions for all other users are set according to the **Children** ACL of the parent folder.

Newly created folders inherit the standard ACLs of the parent folder. They do not inherit the **Children** ACL.

For example, if the **Children** setting of the parent folder's ACL includes Full Control permission for the Administrator and View permission for the Everyone group, then the newly created object inside that folder will have Full Control permission for the owner, Full Control for the Administrator, and View permission for Everyone.

Modifying the ACL of a shortcut object does not modify the ACL of that shortcut's parent object.

When you move an object to a different folder, the moved object retains its original ACLs. When you copy an object, the copied object inherits its ACL from the **Children** ACL of the folder into which it is copied.

What permissions can be granted for an object?

When you edit an object's ACL using the object's Properties dialog box, you can assign a predefined grouping of permissions or you can create a custom grouping. The table below lists the predefined groupings and the specific permissions each one grants.

Grouping	Description	Permissions granted
View	Grants permission to access the object for viewing only, and to provide translations for an object's name and description.	• Browse • Read • Use • Execute

 © 2016, MicroStrategy Inc.

Grouping	Description	Permissions granted
Modify	Grants permission to view and/or modify the object.	• Browse • Read • Write • Delete • Use • Execute
Full Control	Grants all permissions for the object and also allows to modify the ACL for the object.	Control and all other permissions are granted
Denied All	Explicitly denies all permissions for the object. None of the permissions are assigned.	none; all are denied
Default	Neither grants nor denies permissions. All permissions are inherited from the groups to which the user or group belongs.	none
Custom	Allows the user or group to have a custom combination of permissions that you can define.	custom choice
Consume (Only available in MicroStrategy Web)	(Intelligent Cube only) Grants permission to create and execute reports based on this Intelligent Cube.	• Browse • Read • Use
Add (Only available in MicroStrategy Web)	(Intelligent Cube only) Grants permission to create and execute reports based on this Intelligent Cube, and republish/re-execute the Intelligent Cube to update the data.	• Browse • Read • Use • Execute
Collaborate (Only available in MicroStrategy Web)	(Intelligent Cube only) Grants permission to create and execute reports based on this Intelligent Cube, republish/re-execute the Intelligent Cube to update the data, and modify the Intelligent Cube.	• Browse • Read • Write • Delete • Use • Execute

The permissions actually assigned to the user or group when you select a permission grouping are explained in the table below.

Permission	Definition
Browse	View the object in Developer and MicroStrategy Web

Permission	Definition
Read	View the object's definition in the appropriate editor, and view the object's access control list. When applied to a language object, allows users to see the language in the Translation Editor but not edit strings for this language.
Write	Modify the object's definition in the appropriate editor and create new objects in the parent object. For example, add a new metric in a report or add a new report to a document.
Delete	Delete the object
Control	Modify the object's access control list
Use	Use the object when creating or modifying other objects. For example, the Use permission on a metric allows a user to create a report containing that metric. For more information, see *Permissions and report/document execution, page 65*. When applied to a language object, allows users to edit and save translations, and to select the language for display in their Developer or MicroStrategy Web language preferences. This permission is checked at design time, and when executing reports against an Intelligent Cube. **Note**: A user with Use but not Execute permission for an Intelligent Cube can create and execute reports that use that Intelligent Cube, but cannot publish the Intelligent Cube.
Execute	Execute reports or documents that reference the object. To execute a report or document, a user must have Execute access to all objects on the report/document. For more information, see *Permissions and report/document execution, page 65*. This permission is checked at run time. **Note**: The user must have Use permission on an Intelligent Cube to execute reports against that Intelligent Cube.

When you give users only Browse access to a folder, using the Custom permissions, they can see that folder displayed, but cannot see a list of objects within the folder. However, if they perform a search, and objects within that folder match the search criteria, they can see those objects. To deny a user the ability to see objects within a folder, you must deny all access directly to the objects in the folder.

For example, grant the Browse permission to a folder, but assign Denied All for the folder's children objects, then select the **Apply changes in permissions to all children objects** check box. This allows a user to see the folder, but nothing inside it. Alternatively, if you assign Denied All to the folder and to its children, the user cannot see the folder or any of its contents.

Permissions for server governing and configuration

A server object is a configuration-level object in the metadata called Server Definition. It contains governing settings that apply at the server level, a list of projects registered on the server, connection information to the metadata repository, and so on. It is created or modified when a user goes through the Configuration Wizard. Server definition objects are not displayed in the interface in the same way other objects are (reports, metrics, and so on).

As with other objects in the system, you can create an ACL for a server object that determines what system administration permissions are assigned to which users. These permissions are different from the ones for other objects (see table below) and determine what capabilities a user has for a specific server. For example, you can configure a user to act as an administrator on one server, but as an ordinary user on another. To do this, you

 © 2016, MicroStrategy Inc.

must modify the ACL for each server definition object by right-clicking the **Administration** icon, selecting **Properties**, and then selecting the **Security** tab.

The table below lists the groupings available for server objects, the permissions each one grants, and the tasks each allows you to perform on the server.

Grouping	Permissions Granted	Allows you to...
Connect	• Browse	Connect to the server
Monitoring	• Browse • Read	• View server definition properties • View statistics settings • Use the system monitors
Administration	• Browse • Read • Use • Execute	• Start/stop the server • Apply runtime settings • Update diagnostics at runtime • Cancel jobs • Idle/resume a project • Disconnect user • Schedule reports • Delete schedules • Trigger events • Perform cache administration • Create security filters • Use Security Filter Manager
Configuration	• Browse • Read • Write • Delete • Control	• Change server definition properties • Change statistics settings • Delete server definition • Grant server rights to other users
Default	All permissions that are assigned to "Default"	Perform any task on that server.
Custom...	custom choice	Perform the tasks your custom selections allow.

How permissions are determined

A user can have permissions for a given object from the following sources:

- User identity: The user identity is what determines an object's owner when an object is created. The user identity also determines whether or not the user has been granted the right to access a given object.
- Group membership: A user is granted access to an object if he or she belongs to a group with access to the object.

- Special privileges: A user may possess a special privilege that causes the normal access checks to be bypassed:
  - Bypass Schema Object Security Access Checks allows the user to ignore the access checks for schema objects.
  - Bypass All Object Security Access Checks allows the user to ignore the access checks for all objects.

Permission levels

A user can have permissions directly assigned to an object, and be a member of one or more groups that have a different permission grouping assigned to the object. In this case, user-level permissions override group-level permissions, and permissions that are denied at the user or group level override permissions that are granted at that level. The list below indicates what permissions are granted when permissions from multiple sources conflict.

1. Permissions that are directly denied to the user are always denied.
2. Permissions that are directly granted to the user, and not directly denied, are always granted.
3. Permissions that are denied by a group, and not directly granted to the user, are denied.
4. Permissions that are granted by a group, and not denied by another group or directly denied, are granted.
5. Any permissions that are not granted, either directly or by a group, are denied.

For example, user Jane does not have any permissions directly assigned for a report. However, Jane is a member of the Designers group, which has Full Control permissions for that report, and is also a member of the Managers group, which has Denied All permissions for that report. In this case, Jane is denied all permissions for the report. If Jane is later directly granted View permissions for the report, she would have View permissions only.

Default permissions for folders in a new project

By default, in a new MicroStrategy project, users are only allowed to save objects within their personal folders. Only administrative users can save objects within the Public Folder directory in a MicroStrategy project. Folders in a new project are created with these default ACLs:

- Public Objects folder, Schema Objects folder
  - Administrator: Full Control
  - Everyone: Browse
  - Public/Guest: Browse
- Inherited ACL

 © 2016, MicroStrategy Inc.

    - Administrator: Default
    - Everyone: View
    - Public/Guest: View

    This means that new users, as part of the Everyone group, are able to browse the objects in the Public Objects folder, view their definitions and use them in definitions of other objects (for example, create a report with a public metric), and execute them (execute reports). However, new users cannot delete these objects, or create or save new objects to these folders.

- Personal folders
    - Owner: Full Control

    This means that new users can create objects in these folders and have full control over those objects.

Permissions and report/document execution

Two permissions relate to report and document execution: the Use and Execute permissions. These have the following effects:

- The **Use** permission allows the user to reference or use the object when they are modifying another object. This permission is checked at object design time, and when executing reports against an Intelligent Cube.
- The **Execute** permission allows the user to execute reports or documents that use the object. This permission is checked only at report/document execution time.

A user may have four different levels of access to an object using these two new permissions:

- Both Use and Execute permissions: The user can use the object to create new reports, and can execute reports containing the object.
- Execute permission only: The user can execute previously created reports containing the object, but cannot create new reports that use the object. If the object is an Intelligent Cube, the user cannot execute reports against that Intelligent Cube.
- Use permission only: The user can create reports using the object, but cannot execute those reports.

    A user with Browse, Read, and Use (but not Execute) permissions for an Intelligent Cube can create and execute reports that use that Intelligent Cube, but cannot publish the Intelligent Cube.

- Neither Use nor Execute permission: The user cannot create reports containing the object, nor can the user execute such reports, even if the user has Execute rights on the report.

© 2016, MicroStrategy Inc.

Interpreting access rights during report/document execution

The ability to execute a report or document is determined by whether the user has Execute permission on the report and Execute permission on the objects used to define that report. More specifically, Execute permission is required on all attributes, custom groups, consolidations, prompts, metrics, facts, filters, templates, and hierarchies used to define the report or document. Permissions are not checked on transformations and functions used to define the report.

If the user does not have access to an attribute, custom group, consolidation, prompt, fact, filter, template, or hierarchy used to define a report, the report execution fails.

If the user does not have access to a metric used to define a report, the report execution continues, but the metric is not displayed in the report for that user.

This enhancement allows a finer level of access control when executing reports. The same report can be deployed to many users who experience different results depending on their respective permissions on metrics.

ACLs and personalized drill paths in MicroStrategy Web

You can control what attribute drill paths users see on reports. You can determine whether users can see all drill paths for an attribute, or only those to which they have access. You determine this access using the **Enable Web personalized drill paths** check box in the Project Configuration Editor, **Project Definition**: **Drilling** category. (In Developer, right-click a project and select **Project Configuration**.)

With the **Enable Web personalized drill paths** check box cleared (and thus, XML caching enabled), the attributes to which all users in MicroStrategy Web can drill are stored in a report's XML cache. In this case, users see all attribute drill paths whether they have access to them or not. When a user selects an attribute drill path, Intelligence Server then checks whether the user has access to the attribute. If the user does not have access (for example, because of Access Control Lists), the drill is not performed and the user sees an error message.

Alternatively, if you select the **Enable Web personalized drill paths** check box, at the time the report results are created (not at drill time), Intelligence Server checks which attributes the user may access and creates the report XML with only the allowed attributes. This way, the users only see their available drill paths, and they cannot attempt a drill action that is not allowed. With this option enabled, you may see performance degradation on Intelligence Server. This is because it must create XML for each report/user combination rather than using XML that was cached.

For more information about XML caching, see *XML caches, page 404*.

Controlling access to functionality: Privileges

As discussed earlier in this chapter, there are different types of users and groups in the user community. It is your responsibility as a system administrator to assign privileges to users and groups. They give you full control over the user experience.

Privileges give users access to specific MicroStrategy functionality. For example, the Create Metric privilege allows the user to use the Metric Editor to create a new metric,

 © 2016, MicroStrategy Inc.

and the Monitor Caches privilege allows the user to view cache information in the Cache Monitor.

There is a special privilege called Bypass All Object Security Access Checks. Users with this privilege can ignore the access control permissions and are considered to have full control over all objects. For information about permissions, see *Controlling access to objects: Permissions, page 58*.

Based on their different privileges, the users and user groups can perform different types of operations in the MicroStrategy system. If a user does not have a certain privilege, that user does not have access to that privilege's functionality. You can see which users are using certain privileges by using License Manager (see *Using License Manager, page 207*).

Most privileges may be granted within a specific project or across all projects. Certain administrative privileges, such as Configure Group Membership, do not apply to specific projects and can only be granted at the project source level.

For a complete list of privileges and what they control in the system, see the *List of Privileges* chapter in the Supplemental Reference for System Administration.

Assigning privileges to users and groups

Privileges can be assigned to users and user groups directly or through security roles. The difference is that the former grants functionality across all projects while the latter only apply within a specified project (see *Defining sets of privileges: Security roles, page 69*).

To assign privileges to users or groups

1. From Developer User Manager, edit the user with the User Editor or edit the group with the Group Editor.
2. Expand **User Definition** or **Group Definition**, and then select **Project Access**.
3. Select the check boxes to grant privileges to the user or group.

Rather than assigning individual users and groups these privileges, it may be easier for you to create Security Roles (collections of privileges) and assign them to users and groups. Then you can assign additional privileges individually when there are exceptions. For more information about security roles, see *Defining sets of privileges: Security roles, page 69*.

Assigning privileges to multiple users at once

You can grant, revoke, and replace the existing privileges of users, user groups, or security roles with the Find and Replace Privileges dialog box. This dialog box allows you to search for the user, user group, or security role and change their privileges, depending on the tasks required for their work.

© 2016, MicroStrategy Inc.

For example, your organization is upgrading Flash on all users' machines. Until the time the Flash update is completed, the users will not be able to export reports to Flash. You can use Find and Replace Privileges to revoke the Export to Flash privilege assigned to users, and when the upgrade is complete you can grant the privilege to the users again.

To access the Find and Replace Privileges dialog box, in Developer, right-click the User Manager and select **Find and Replace Privileges**. The Find and Replace Privileges dialog box opens. For detailed instructions on how to find and replace privileges, see the *MicroStrategy Developer Help*.

How are privileges inherited?

A user's privileges within a given project include the following:

- Privileges assigned directly to the user (see *Assigning privileges to users and groups, page 67*)
- Privileges assigned to any groups of which the user is a member (see *About MicroStrategy user groups, page 52*)

Groups also inherit privileges from their parent groups.

- Privileges assigned to any security roles that are assigned to the user within the project (see *Defining sets of privileges: Security roles, page 69*)
- Privileges assigned to any security roles that are assigned to a group of which the user is a member

Predefined user groups and privileges

MicroStrategy comes with several predefined user groups. For a complete list and explanation of these groups, see *About MicroStrategy user groups, page 52*. These groups possess the following privileges:

- Everyone, Public/Guest, Third Party Users, LDAP Public/Guest, and LDAP Users, have no predefined privileges.
- The predefined product-based user groups possess all the privileges associated with their corresponding products. For a list of these groups, see *Groups corresponding to product offerings, page 54*.

International Users is a member of the following product-based groups: Analyst, Mobile User, Web Reporter, and Web Analyst. It has the privileges associated with these groups.

- System Monitors and its member groups have privileges based on their expected roles in the company. To see the privileges assigned to each group, right-click the group and select **Grant Access to Projects**.

 © 2016, MicroStrategy Inc.

How predefined user groups inherit privileges

Several of the predefined user groups form hierarchies, which allow groups to inherit privileges from any groups at a higher level within the hierarchy. These hierarchies are as follows:

- Web Reporter
    - Web Analyst
        - Web Professional

    In the case of the MicroStrategy Web user groups, the Web Analyst inherits the privileges of the Web Reporter. The Web Professional inherits the privileges of both the Web Analyst and Web Reporter. The Web Professional user group has the complete set of MicroStrategy Web privileges.

- Analyst
    - Developer

    In the case of the MicroStrategy Developer user groups, the Developer inherits the privileges of the Analyst and therefore has more privileges than the Analysts.

- System Monitors
    - various System Monitors groups

    The various System Monitors user groups inherit the privileges of the System Monitors user group and therefore have more privileges than the System Monitors. Each has its own specific set of privileges in addition, that are not shared by the other System Monitors groups.

- International Users

    This group inherits the privileges of the Analyst, Mobile User, Web Reporter, and Web Analyst groups.

Defining sets of privileges: Security roles

A security role is a collection of project-level privileges that are assigned to users and groups. For example, you might have two types of users with different functionality needs: the Executive Users who need to run, sort, and print reports, and the Business Analysts who need additional capabilities to drill and change subtotal definitions. In this case, you can create two security roles to suit these two different types of users.

Security roles exist at the project source level, and can be used in any project registered with Intelligence Server. A user can have different security roles in each project. For example, an administrator for the development project may have a Project Administrator security role in that project, but the Normal User security role in all other projects on that server.

A security role is fundamentally different from a user group in the following ways:

- A group is a collection of users that can be assigned privileges (or security roles) all at once, for the project source and all projects in it.
- A security role is a collection of privileges in a project. Those privileges are assigned as a set to various users or groups, on a project-by-project basis.

For information about how privileges are inherited from security roles and groups, see *How are privileges inherited?, page 68*.

Managing security roles

The Security Role Manager lists all the security roles available in a project source. From this manager you can assign or revoke security roles for users in projects, or create or delete security roles. For additional methods of managing security roles, see *Other ways of managing security roles, page 71*.

To assign a security role to users or groups in a project

1. In Developer, log in to the project source containing the security role. You must have the Grant/Revoke Privileges privilege.
2. Expand **Administration**, then **Configuration Managers**, and then select **Security Roles**. A list of security roles in the project source opens in the main Developer pane.
3. Double-click the security role you want to assign to the user or group. The Security Role Editor opens.
4. Select the **Members** tab.
5. From the **Select a Project** drop-down list, select the project for which to assign the security role.
6. From the drop-down list of groups, select the group containing a user or group you want to assign the security role to. The users or groups that are members of that group are shown in the list box below the drop-down list.

   Note the following:

   — By default, users are not shown in this list box. To view the users as well as the groups, select the **Show users** check box.

   — To assign a top-level group to a security role, from the drop-down list select **All Groups**.
7. Select a desired user or group.
8. Click the **>** icon. The user or group moves to the **Selected members** list. You can assign multiple users or groups to the security role by selecting them and clicking the **>** icon.
9. When you are finished assigning the security role, click **OK**. The security role is assigned to the selected users and groups and the Security Role Editor closes.

 © 2016, MicroStrategy Inc.

To create a security role

1. In Developer, log in to a project in the project source you want to create the security role in.
2. Expand **Administration**, then **Configuration Managers**, and then select **Security Roles**.
3. From the **File** menu, point to **New**, and select **Security Role**. The Security Role Editor opens at the **General** tab.
4. Enter a name and description for the new security role.
5. Select the **Privileges** tab.
6. Select the privileges to add to this security role. For an explanation of each privilege, see the *List of Privileges* chapter in the Supplemental Reference for System Administration.

To select all privileges in a privilege group, select the group.

7. To assign the role to users, select the **Members** tab and follow the instructions in *To assign a security role to users or groups in a project, page 70*.
8. Click **OK** to close the Security Role Editor and create the security role.

Other ways of managing security roles

You can also assign security roles to a user or group in the User Editor or Group Editor. From the **Project Access** category of the editor, you can specify what security roles that user or group has for each project.

You can assign roles to multiple users and groups in a project through the Project Configuration dialog box. The **Project Access - General** category displays which users and groups have which security roles in the project, and allows you to re-assign the security roles.

For detailed instructions on using these editors to manage security roles, see the *MicroStrategy Developer Help*.

You can also use Command Manager to manage security roles. Command Manager is a script-based administrative tool that helps you perform complex administrative actions quickly. For specific syntax for security role management statements in Command Manager, see Security Role Management in the Command Manager on-line help (from Command Manager, press **F1**, or select the **Help** menu). For general information about Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*.

If you are using UNIX, you must use Command Manager to manage your system's security roles.

Controlling access to a project

You can deny user or group access to a specific MicroStrategy project by using a security role.

To deny user or group access to a project

1. In Developer, right-click on the project you want to deny access to. Select **Project Configuration**. The Project Configuration Editor opens.
2. Expand the **Project Access** category. The Project Access - General dialog box opens.
3. In the **Select a security role** drop-down list, select the security role that contains the user or group who you want to deny project access. For example, select the Normal Users security role.
4. On the right-hand side of the Project access - General dialog, select the user or group who you want to deny project access. Then click the left arrow to remove that user or group from the security role. For example, remove the Everyone group.
5. Using the right arrow, add any users to the security role for whom you want to grant project access. To see the users contained in each group, highlight the group and check the **Show users** check box.
6. Make sure the user or group whose access you want deny does not appear in the **Selected members** pane on the right-hand side of the dialog. Then click **OK**.
7. In Developer, under the project source that contains the project you are restricting access to, expand **Administration**, then expand **User Manager**.
8. Click on the group to which the user belongs who you want to deny project access for. Then double-click on the user in the right-hand side of Developer. The User Editor opens.
9. Expand User Definition, then select Project Access.
10. In the Security Role Selection row, under the project you want to restrict access to, review the Security Role Selection drop-down list. Make sure that no security role is associated with this project for this user.
11. Click **OK**.

When the user attempts to log in to the project, he receives the message “No projects were returned by this project source.”

The role-based administration model

Beginning with version 9.0, the MicroStrategy product suite comes with a number of predefined security roles for administrators. These roles makes it easy to delegate administrative tasks.

 © 2016, MicroStrategy Inc.

For example, your company security policy may require you to keep the user security administrator for your projects separate from the project resource administrator. Rather than specifying the privileges for each administrator individually, you can assign the Project Security Administrator role to one administrator, and the Project Resource Administrator to another. Because users can have different security roles for each project, you can use the same security role for different users in different projects to further delegate project administration duties.

The predefined project administration roles cover every project-level administrative privilege except for Bypass All Object Security Access Checks. None of the roles have any privileges in common. For a list of the privileges included with each predefined security role, see the *List of Privileges* chapter in the *Supplemental Reference for System Administration*.

The predefined administration security roles are:

- **Power Users**, which have the largest subset of privileges of any security role.
- **Project Bulk Administrators**, who can perform administrative functions on multiple objects with Object Manager (see *Copying objects between projects: Object Manager, page 232*), Command Manager (see *Chapter 15, Automating Administrative Tasks with Command Manager*), and the Bulk Repository Translation Tool.
- **Project Operations Administrators**, who can perform maintenance on various aspects of a project.
- **Project Operations Monitors**, who can view the various Intelligence Server monitors but cannot make any changes to the monitored systems.
- **Project Resource Settings Administrators**, who can configure project-level settings.
- **Project Security Administrators**, who create users and manage user and object security.

For instructions on how to assign these security roles to users or groups, see *Managing security roles, page 70*.

Do not modify the privileges for an out-of-the-box security role. During upgrades to newer versions of MicroStrategy, the privileges for the out-of-the-box security roles are overwritten with the default privileges. Instead, you should copy the security role you need to modify and make changes to the copied version.

Controlling access to data

Access control governs the resources that an authenticated user is able to read, modify, or write. Data is a major resource of interest in any security scheme that determines what source data a user is allowed to access. You may be more familiar with the terms *authentication* (making sure the user is who he says he is) and *authorization* (making sure he can access the data he is entitled to see now that I know who he is).

© 2016, MicroStrategy Inc.

The ways by which data access can be controlled are discussed below:

- *Controlling access to the database: Connection mappings, page 74*
- *Linking database users and MicroStrategy users: Passthrough execution, page 77*
- *Restricting access to data: Security filters, page 79*
- *Controlling access to data at the database (RDBMS) level, page 91*

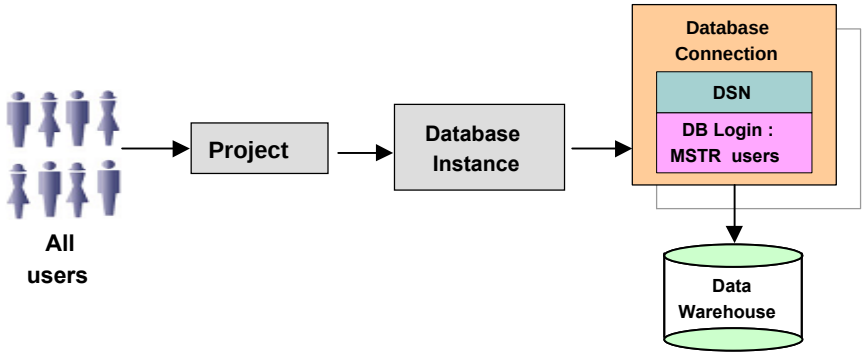
Controlling access to the database: Connection mappings

Connection mappings allow you to assign a user or group in the MicroStrategy system to a login ID on the data warehouse RDBMS. The mappings are typically used to take advantage of one of several RDBMS data security techniques (security views, split fact tables by rows, split fact tables by columns) that you may have already created. For details on these techniques, see *Controlling access to data at the database (RDBMS) level, page 91*.

Why use connection mappings?

Use a connection mapping if you need to differentiate MicroStrategy users from each other at the data warehouse level or if you need to direct them to separate data warehouses. This is explained in more detail below.

First it is important to know that, as a default, all users in a MicroStrategy project use the same database connection/DSN and database login when connecting to the database. This means that all users have the same security level at the data warehouse and therefore, security views cannot be assigned to a specific MicroStrategy user. In this default configuration, when the database administrator (DBA) uses an RDBMS feature to view a list of users connected to the data warehouse, all MicroStrategy users would all appear with the same name. For example, if forty users are signed on to the MicroStrategy system and running jobs, the DBA sees a list of forty users called “MSTR users” (or whatever name is specified in the default database login). This is shown in the diagram below in which all jobs running against the data warehouse use the “MSTR users” database login.



 © 2016, MicroStrategy Inc.

Creating a connection mapping

You define connection mappings with the Project Configuration Editor in Developer. To create a connection mapping, you assign a user or group either a database connection or database login that is different from the default. For information on this, see *Connecting to the data warehouse, page 8*.

To create a connection mapping

1. In Developer, log into your project. You must log in as a user with administrative privileges.
2. From the **Administration** menu, point to **Projects**, and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand the **Database Instances** category, and then select **Connection Mapping**.
4. Right-click in the grid and select **New** to create a new connection mapping.
5. Double-click the new connection mapping in each column to select the database instance, database connection, database login, and language.
6. Double-click the new connection mapping in the **Users** column. Click **...** (the browse button). The Add Members dialog box opens.
7. Select the desired user or group and click **OK**. That user or group is now associated with the connection mapping.
8. Click **OK** to close the Project Configuration Editor. The new connection mapping is saved.

Connection mapping example

One case in which you may wish to use connection mappings is if you have existing security views defined in the data warehouse and you wish to allow MicroStrategy users' jobs to execute on the data warehouse using those specific login IDs. For example,

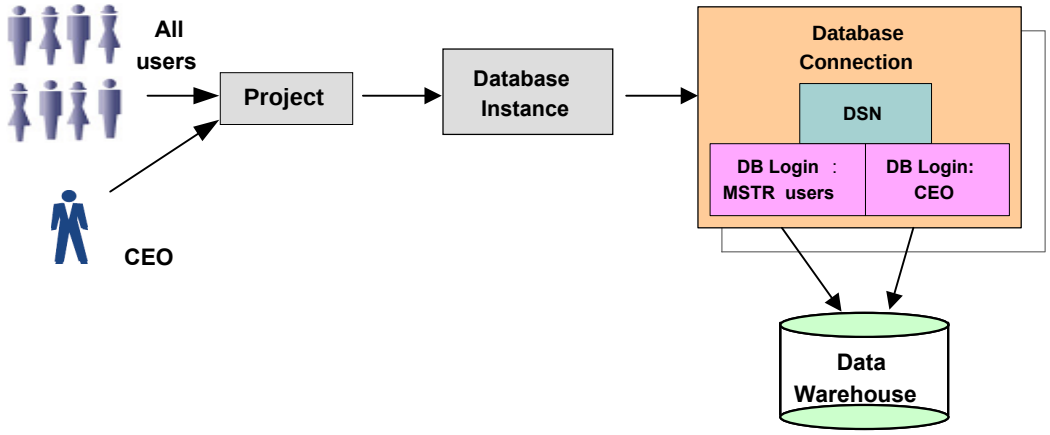
- The CEO can access all data (warehouse login ID = "CEO")
- All other users have limited access (warehouse login ID = "MSTR users")

In this case, you would need to create a user connection mapping within MicroStrategy for the CEO. To do this:

- Create a new database login definition for the CEO in MicroStrategy so it matches his or her existing login ID on the data warehouse
- Create the new connection mapping in MicroStrategy to specify that the CEO user uses the new database login

This is shown in the diagram below in which the CEO connects as CEO (using the new database login called "CEO") and all other users use the default database login "MSTR users."

© 2016, MicroStrategy Inc.



Both the CEO and all the other users use the same project, database instance, database connection (and DSN), but the database login is different for the CEO.

If we were to create a connection mapping in the MicroStrategy Tutorial project according to this example, it would look like the diagram below.

For information on creating a new database connection, see *Connecting to the data warehouse, page 8*. For information on creating a new database login, see *Connecting to the data warehouse, page 8*.

Connection mappings can also be made for user groups and are not limited to individual users. Continuing the example above, if you have a Managers group within the MicroStrategy system that can access most data in the data warehouse (warehouse login ID = “Managers”), you could create another database login and then create another connection mapping to assign it to the Managers user group.

Another case in which you may want to use connection mappings is if you need to have users connect to two data warehouses using the same project. In this case, both data warehouses must have the same structure so that the project works with both. This may be applicable if you have a data warehouse with domestic data and another with foreign data and you want users to be directed to one or the other based on the user group to which they belong when they log in to the MicroStrategy system.

For example, if you have two user groups such that:

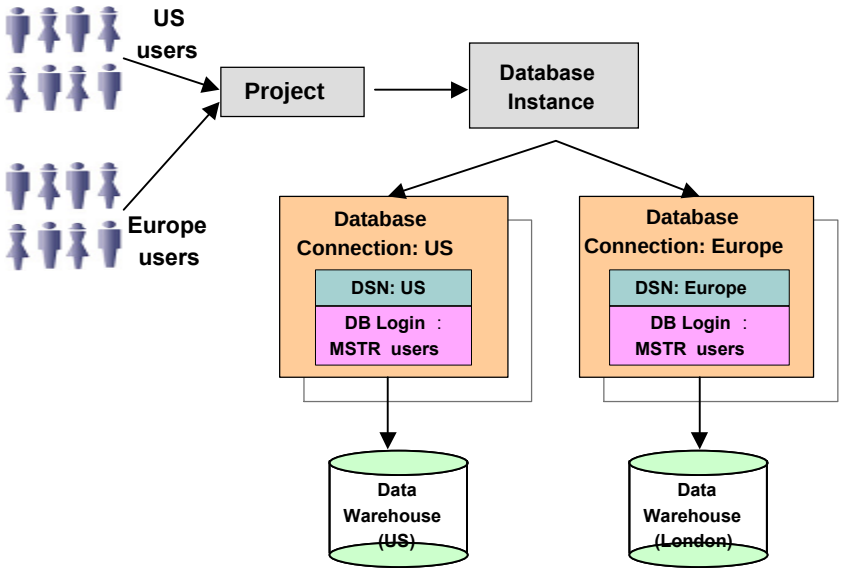
 © 2016, MicroStrategy Inc.

- "US users" connect to the U.S. data warehouse (data warehouse login ID "MSTR users")
- "Europe users" connect to the London data warehouse (data warehouse login ID "MSTR users")

In this case, you would need to create a user connection mapping within MicroStrategy for both user groups. To do this, you would:

- Create two database connections in MicroStrategy—one to each data warehouse (this assumes that DSNs already exist for each data warehouse)
- Create two connection mappings in the MicroStrategy project that link the groups to the different data warehouses via the two new database connection definitions

This is shown in the diagram below.



The project, database instance, and database login can be the same, but the connection mapping specifies different database connections (and therefore, different DSNs) for the two groups.

Linking database users and MicroStrategy users: Passthrough execution

You can link a MicroStrategy user to an RDBMS login ID using the User Editor (on the Authentication tab, specify the **Warehouse Login** and **Password**) or using Command Manager. This link is required for database warehouse authentication (see *Implementing database warehouse authentication, page 103*) but works for other authentication modes as well.

© 2016, MicroStrategy Inc.

You can configure each project to use either connection mappings or the linked warehouse login ID when users execute reports, documents, or browse attribute elements. If passthrough execution is enabled, the project uses the linked warehouse login ID and password as defined in the User Editor (Authentication tab). If no warehouse login ID is linked to a user, Intelligence Server uses the default connection and login ID for the project's database instance.

By default, warehouse passthrough execution is turned off, and the system uses connection mappings. If no connection mapping is defined for the user, Intelligence Server uses the default connection and login ID for the project's database instance.

Why use passthrough execution?

You may wish to use passthrough execution for these reasons:

- RDBMS auditing: If you wish to be able to track which users are accessing the RDBMS system down to the individual database query. Mapping multiple users to the same RDBMS account blurs the ability to track which users have issued which RDBMS queries.
- Teradata spool space: If you use the Teradata RDBMS, note that it has a limit for spool space set per account. If multiple users share the same RDBMS account, they are collectively limited by this setting.
- RDBMS security views: If you use security views, each user needs to log in to the RDBMS with a unique database login ID so that a database security view is enforced.

Enabling linked warehouse logins

You can configure linked warehouse logins with the Project Configuration Editor in Developer. To create a connection mapping, you assign a user or group either a database connection or database login that is different from the default. For information on this, see *Connecting to the data warehouse, page 8*.

To enable linked warehouse logins

1. In Developer, log into your project. You must log in as a user with administrative privileges.
2. From the **Administration** menu, point to **Projects**, and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand the **Database Instances** category, expand **Authentication**, and then select **Warehouse**.
4. Select the **Use warehouse pass-through credentials** check box.
5. To use warehouse credentials for all database instances, select the **For all database instances** option.

© 2016, MicroStrategy Inc.

6 To use warehouse credentials for specific database instances, select the **For selected database instances** option. Then select those database instances from the list below.

7 Click **OK**. The Project Configuration Editor closes and the changes are saved.

Restricting access to data: Security filters

Security filters enable you to control what warehouse data users can see when that data is accessed through MicroStrategy. A security filter can be assigned to a user or group to narrow the result set when they execute reports or browse elements. The security filter applies to all reports and documents, and all attribute element requests, submitted by a user.

For example, two regional managers can have two different security filters assigned to them for their regions: one has a security filter assigned to her that only shows the data from the Northeast region, and the other has a security filter that only shows data from the Southwest region. If these two regional managers run the same report, they may see different report results.

Security filters serve a similar function to database-level techniques such as database views and row level security. For information about controlling data security at the data warehouse level, see *Controlling access to data at the database (RDBMS) level, page 91*.

For more information about security filters, see the following:

- *Security filter example, page 79*
- *How security filters work, page 80*
- *Creating and applying a security filter, page 81*
- *Security filters and metric levels, page 82*
- *Using a single security filter for multiple users: System prompts, page 89*
- *Merging security filters, page 86*

Security filter example

A user in the MicroStrategy Tutorial project has a security filter defined as Subcategory=TV. When this user browses the Product hierarchy beginning with the Category attribute, she only sees the Electronics category. Within the Electronics category, she sees only the TV subcategory. Within the TV subcategory, she sees all Items within that subcategory.

When this user executes a simple report with Category, Subcategory, and Item in the rows, and Revenue in the columns, only the Items from the TV Subcategory are returned, as shown in the example below.

© 2016, MicroStrategy Inc.

Category	Subcategory	Item	Metrics	Revenue
Electronics	TV's	GPX 5" AM/FM Portable TV		$76,703
		RCA 32" Stereo TV		$436,830
		RCA Indoor TV Antenna		$32,781
		RCA Power TV Antenna		$73,040
		RCA 27" Stereo TV		$236,160
		RCA 13" TV/VCR		$188,688
		RCA 13" TV		$92,307
		RCA 4" LCD Color TV		$156,090
		RCA 2" Diagonal LCD Color TV		$145,235
		Sharp 25" TV/VCR Combo		$271,852
		Sharp 25" Stereo Color TV		$208,474
		Sharp 13" 2-Head TV/VCR		$168,947
		Sharp 32" Color TV		$460,395
		Sony 32" Trinitron Television		$586,960
		Sony 35" Trinitron Television		$703,446

If this user executes another report with Category in the rows and Revenue in the columns, only the Revenue from the TV Subcategory is returned, as shown in the example below. The user cannot see any data from attribute elements that are outside the security filter.

Category	Metrics	Revenue
Electronics		$3,837,906

How security filters work

Security filters are the same as regular filters except that they can contain only attribute qualifications, custom expressions, and joint element lists. Relationship filters and metric qualifications are not allowed in a security filter. A security filter can include as many expressions as you need, joined together by logical operators. For more information on creating filters, see the *Filters* chapter in the Basic Reporting Guide.

A security filter comes into play when a user is executing reports and browsing elements. The qualification defined by the security filter is used in the `WHERE` clause for any report that is related to the security filter's attribute. By default, this is also true for element browsing: when a user browses through a hierarchy to answer a prompt, she only sees the attribute elements that the security filter allows her to see. For instructions on how to disable security filters for element browsing, see *To disable security filters for element browsing, page 82*.

Security filters are used as part of the cache key for report caching and element caching. This means that users with different security filters cannot access the same cached results, preserving data security. For more information about caching, see *Chapter 10, Improving Report and Document Response Time: Caching*.

Each user or group can be directly assigned only one security filter for a project. Users and groups can be assigned different security filters for different projects. In cases where a user inherits one or more security filters from any groups that she belongs to, the

 © 2016, MicroStrategy Inc.

security filters may need to be merged. For information about how security filters are merged, see *Merging security filters, page 86*.

Creating and applying a security filter

You create and apply security filters in the Security Filter Manager. Make sure you inform your users of any security filters assigned to them or their group. If you do not inform them of their security filters, they may not know that the data they see in their reports has been filtered, which may cause misinterpretation of report results.

Prerequisites

To create security filters, you must have the following privileges:

- Create Application Objects (under the Common Privileges privilege group)
- Use Report Filter Editor (under the Developer privilege group)
- Use Security Filter Manager (under the Administration privilege group)

To create and apply a security filter for a user or group

1. In Developer, from the **Administration** menu, point to **Projects** and then select **Security Filter Manager**. The Security Filter Manager opens.
2. From the **Choose a project** drop-down list, select the project that you want to create a security filter for.

Create a security filter

3. Select the **Security Filters** tab.
4. Select one:
   - To create a new security filter, click **New**. The Security Filter Editor opens. For instructions on how to use this editor to create a filter, see the *MicroStrategy Developer Help*.
   - OR, to convert an existing filter into a security filter, click **Import**. Browse to the filter you want to convert and click **Open**. Specify a name and location for the new security filter and click **Save**.

Apply the security filter to a user or group

5. In the left side of the Security Filter Manager, in the **Security Filters** tab, browse to the security filter that you want to apply, and select that security filter.
6. In the right side of the Security Filter Manager, select **Security Filters**.
7. Browse to the user or group that you want to apply the security filter to, and select that user or group.

© 2016, MicroStrategy Inc.

8 Click **>** to apply the selected security filter to the selected user or group.

9 Click **OK** to close the Security Filter Manager.

To disable security filters for element browsing

1 In Developer, log into a project. You must log in with a user account that has administrative privileges.

2 From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.

3 Expand the **Project Definition** category, and then select **Advanced**.

4 Under **Attribute element browsing**, clear the **Apply security filters to element browsing** check box.

5 Click **OK** to close the Project Configuration Editor.

6 Restart Intelligence Server for your changes to take effect.

Security filters and metric levels

In certain situations involving level metrics, users may be able to see a limited amount of data from outside their security filter. Specifically, if a metric is defined with absolute filtering on a level above that used in the security filter's expression, the filter expression is raised to the metric's level. For information about metric levels and filtering in metrics, see the *Metrics* chapter in the Advanced Reporting Guide.

For example, consider a metric called Category Revenue that is defined to return the revenue across all items in each category. Its level expression is Target=Category, Filtering=Absolute. When a user with a security filter Subcategory=TV executes a report with the Category Revenue metric, the Category Revenue metric displays the total revenue for the category. The user's security filter is effectively changed to show the entire Category in which TV is a Subcategory.

This behavior can be modified by using the top range attribute and bottom range attribute properties.

- A **top range attribute** specifies the highest level of detail in a given hierarchy that the security filter allows the user to view. If a top range attribute is specified, the security filter expression is not raised to any level above the top range.
- A **bottom range attribute** specifies the lowest level of detail in a given hierarchy that the security filter allows the user to view. If this is not specified, the security filter can view every level lower than the specified top range attribute, as long as it is within the qualification defined by the filter expression.

The top and bottom range attributes can be set to the same level.

For instructions on how to assign range attributes to security filters, see *Assigning a top or bottom range attribute to a security filter, page 85*.

 © 2016, MicroStrategy Inc.

The examples below use a report with Category, Subcategory, and Item on the rows, and three metrics in the columns:

- Revenue
- Subcategory Revenue, which is defined with absolute filtering to the Subcategory level
- Category Revenue, which is defined with absolute filtering to the Category level

The user executing this report has a security filter that restricts the Subcategory to the TV element.

No top or bottom range attribute

If no top or bottom range attribute is specified, then at the level of the security filter (Subcategory) and below, the user cannot see data outside his or her security filter. Above the level of the security filter, the user can see data outside the security filter if it is in a metric with absolute filtering for that level. Even in this case, the user sees only data for the Category in which his or her security filter is defined.

In the example report below, the user's security filter does not specify a top or bottom range attribute. Item-level detail is displayed for only the items within the TV category. The Subcategory Revenue is displayed for all items within the TV subcategory. The Category Revenue is displayed for all items in the Category, including items that are not part of the TV subcategory. However, only the Electronics category is displayed. This illustrates how the security filter Subcategory=TV is raised to the category level such that Category=Electronics is the filter used with Category Revenue.

Category	Subcategory	Item	Metrics	Revenue	Category Revenue	Subcategory Revenue
Electronics	TV's	GPX 5" AM/FM Portable TV		$76,703	$24,391,302.85	$3,837,905.85
		RCA 32" Stereo TV		$436,830	$24,391,302.85	$3,837,905.85
		RCA Indoor TV Antenna		$32,781	$24,391,302.85	$3,837,905.85
		RCA Power TV Antenna		$73,040	$24,391,302.85	$3,837,905.85
		RCA 27" Stereo TV		$236,160	$24,391,302.85	$3,837,905.85
		RCA 13" TV/VCR		$188,688	$24,391,302.85	$3,837,905.85
		RCA 13" TV		$92,307	$24,391,302.85	$3,837,905.85
		RCA 4" LCD Color TV		$156,090	$24,391,302.85	$3,837,905.85
		RCA 2" Diagonal LCD Color TV		$145,235	$24,391,302.85	$3,837,905.85
		Sharp 25" TV/VCR Combo		$271,852	$24,391,302.85	$3,837,905.85
		Sharp 25" Stereo Color TV		$208,474	$24,391,302.85	$3,837,905.85
		Sharp 13" 2-Head TV/VCR		$168,947	$24,391,302.85	$3,837,905.85
		Sharp 32" Color TV		$460,395	$24,391,302.85	$3,837,905.85
		Sony 32" Trinitron Television		$586,960	$24,391,302.85	$3,837,905.85
		Sony 35" Trinitron Television		$703,446	$24,391,302.85	$3,837,905.85

Top range attribute: Subcategory

If a top range attribute is specified, then the user cannot see any data outside of her security filter. This is true even at levels above the top level, regardless of whether

metrics with absolute filtering are used.

In the example report below, the user's security filter specifies a top range attribute of Subcategory. Here, the Category Revenue is displayed for only the items within the TV subcategory. The security filter Subcategory=TV is not raised to the Category level, because Category is above the specified top level of Subcategory.

Category	Subcategory	Item	Metrics	Revenue	Category Revenue	Subcategory Revenue
Electronics	TV's	GPX 5" AM/FM Portable TV		$76,703	$3,837,905.85	$3,837,905.85
		RCA 32" Stereo TV		$436,830	$3,837,905.85	$3,837,905.85
		RCA Indoor TV Antenna		$32,781	$3,837,905.85	$3,837,905.85
		RCA Power TV Antenna		$73,040	$3,837,905.85	$3,837,905.85
		RCA 27" Stereo TV		$236,160	$3,837,905.85	$3,837,905.85
		RCA 13" TV/VCR		$188,688	$3,837,905.85	$3,837,905.85
		RCA 13" TV		$92,307	$3,837,905.85	$3,837,905.85
		RCA 4" LCD Color TV		$156,090	$3,837,905.85	$3,837,905.85
		RCA 2" Diagonal LCD Color TV		$145,235	$3,837,905.85	$3,837,905.85
		Sharp 25" TV/VCR Combo		$271,852	$3,837,905.85	$3,837,905.85
		Sharp 25" Stereo Color TV		$208,474	$3,837,905.85	$3,837,905.85
		Sharp 13" 2-Head TV/VCR		$168,947	$3,837,905.85	$3,837,905.85
		Sharp 32" Color TV		$460,395	$3,837,905.85	$3,837,905.85
		Sony 32" Trinitron Television		$586,960	$3,837,905.85	$3,837,905.85
		Sony 35" Trinitron Television		$703,446	$3,837,905.85	$3,837,905.85

Bottom range attribute: Subcategory

If a bottom range attribute is specified, the user cannot see data aggregated at a lower level than the bottom level.

In the example report below, the user's security filter specifies a bottom range attribute of Subcategory. Item-level detail is not displayed, because Item is a level below the bottom level of Subcategory. Instead, data for the entire Subcategory is shown for each item. Data at the Subcategory level is essentially the lowest level of granularity the user is allowed to see.

 © 2016, MicroStrategy Inc.

Category	Subcategory	Item	Metrics	Revenue	Category Revenue	Subcategory Revenue
Electronics	TV's	GPX 5" AM/FM Portable TV		$3,837,906	$24,391,302.85	$3,837,905.85
		RCA 32" Stereo TV		$3,837,906	$24,391,302.85	$3,837,905.85
		RCA Indoor TV Antenna		$3,837,906	$24,391,302.85	$3,837,905.85
		RCA Power TV Antenna		$3,837,906	$24,391,302.85	$3,837,905.85
		RCA 27" Stereo TV		$3,837,906	$24,391,302.85	$3,837,905.85
		RCA 13" TV/VCR		$3,837,906	$24,391,302.85	$3,837,905.85
		RCA 13" TV		$3,837,906	$24,391,302.85	$3,837,905.85
		RCA 4" LCD Color TV		$3,837,906	$24,391,302.85	$3,837,905.85
		RCA 2" Diagonal LCD Color TV		$3,837,906	$24,391,302.85	$3,837,905.85
		Sharp 25" TV/VCR Combo		$3,837,906	$24,391,302.85	$3,837,905.85
		Sharp 25" Stereo Color TV		$3,837,906	$24,391,302.85	$3,837,905.85
		Sharp 13" 2-Head TV/VCR		$3,837,906	$24,391,302.85	$3,837,905.85
		Sharp 32" Color TV		$3,837,906	$24,391,302.85	$3,837,905.85
		Sony 32" Trinitron Television		$3,837,906	$24,391,302.85	$3,837,905.85
		Sony 35" Trinitron Television		$3,837,906	$24,391,302.85	$3,837,905.85

Assigning a top or bottom range attribute to a security filter

You assign top and bottom range attributes to security filters in the Security Filter Manager. You can assign range attributes to a security filter for all users, or to the security filters per user.

You can assign the same attribute to a security filter as a top and bottom range attribute. A security filter can have multiple top or bottom range attributes as long as they are from different hierarchies. You cannot assign multiple attributes from the same hierarchy to either a top or bottom range. However, you can assign attributes from the same hierarchy if one is a top range attribute and one is a bottom range attribute. For example, you can assign Quarter (from the Time hierarchy) and Subcategory (from the Products hierarchy) as top range attributes, and Month (from the Time hierarchy) and Subcategory as bottom range attributes.

Prerequisites

To modify security filters, you must have the Use Security Filter Manager privilege.

To assign a top or bottom range attribute to a security filter

1. In Developer, from the **Administration** menu, point to **Projects** and then select **Security Filter Manager**. The Security Filter Manager opens.
2. From the **Choose a project** drop-down list, select the project that you want to modify security filters for.
3. Select the **Attributes** tab.
4. Browse to the attribute that you want to set as a top or bottom range attribute, and select that attribute.

5 To apply a top or bottom range attribute to a security filter for all users:

   a In the right side of the Security Filter Manager, select **Security Filters**.

   b Browse to the security filter that you want to apply the range attribute to.

   c Expand that security filter, and select either the **Top range attributes** or **Bottom range attributes** folder.

   d Click **>** to apply the selected attribute to the selected security filter.

6 To apply a top or bottom range attribute to a security filter for a single user or group:

   a In the right side of the Security Filter Manager, select **Groups/Users**.

   b Browse to the user or group that you want to apply the range attribute to.

   c Expand that user or group and select the security filter that you want to apply the range attribute to.

   d Expand that security filter, and select either the **Top range attributes** or **Bottom range attributes** folder.

   e Click **>** to apply the selected attribute to the selected security filter for the selected user or group.

7 Click **OK** to close the Security Filter Manager.

Merging security filters

A user can be assigned a security filter directly, and can inherit a security filter from any groups that she belongs to. Because of this, multiple security filters may need to be merged when executing reports or browsing elements.

MicroStrategy supports the following methods of merging security filters:

- *Merging related security filters with OR and unrelated security filters with AND, page 87* (This is the default method for merging security filters)
- *Merging all security filters with AND, page 88*

For the examples in these sections, consider a project with the following user groups and associated security filters:

Group	Security Filter	Hierarchy
Electronics	Category = Electronics	Product
Drama	Subcategory = Drama	Product
Movies	Category = Movies	Product
Northeast	Region = Northeast	Geography

You control how security filters are merged at the project level. You can change the merge settings in the Project Configuration Editor for the selected project, in the

 © 2016, MicroStrategy Inc.

Security Filter category. After making any changes to the security filter settings, you must restart Intelligence Server for those changes to take effect.

Changing how security filters are merged does not automatically invalidate any result caches created for users who have multiple security filters. MicroStrategy recommends that you invalidate all result caches in a project after changing how security filters are merged for that project. For instructions on how to invalidate all result caches in a project, see *Invalidating result caches, page 413*.

Merging related security filters with OR and unrelated security filters with AND

By default, security filters are merged with an OR if they are related, and with an AND if they are not related. That is, if two security filters are related, the user can see all data available from either security filter. However, if the security filters are not related, the user can see only the data available in both security filters.

Two security filters are considered related if the attributes that they derive from belong in the same hierarchy, such as Country and Region, or Year and Month. In the example security filters given above, the Electronics, TV, and Movies security filters are all related, and the Northeast security filter is not related to any of the others.

Using this merge method, a user who is a member of both the Electronics and Drama groups can see data from the Electronics category and the Drama subcategory, as shown below:

Category	Subcategory	Metrics	Revenue
Electronics	Audio Equipment		$3,782,832
	Cameras		$5,061,148
	Computers		$1,928,998
	Electronics - Miscellaneous		$4,671,957
	TV's		$3,837,906
	Video Equipment		$5,108,464
Movies	Drama		$698,840

A user who is a member of both the Movies and Drama groups can see data from all subcategories in the Movies category, not just the Drama subcategory. A user who is a member of both the Electronics and Drama categories can see data from both categories.

If a user who is a member of the Movies and Northeast groups executes a report with Region, Category, and Subcategory in the rows, only data from the Movies category in the Northeast region is shown, as seen below:

Region	Category	Subcategory	Metrics	Revenue
Northeast	Movies	Action		$148,648
		Comedy		$164,441
		Drama		$168,547
		Horror		$155,926
		Kids / Family		$162,235
		Special Interests		$201,764

© 2016, MicroStrategy Inc.

Data for the Movies category from outside the Northeast region is not available to this user, nor is data for the Northeast region for other categories.

Merging all security filters with AND

You can also configure Intelligence Server to always merge security filters with an AND, regardless of whether they are related.

As in the first method, a user who is a member of both the Movies and Northeast groups would see only information about the Movies category in the Northeast region.

A user who is a member of both the Movies and Drama groups would see only data from the Drama subcategory of Movies, as shown below:

Category	Subcategory	Metrics	Revenue
Movies	Drama		$698,840

Data for the other subcategories of Drama is not available to this user.

This setting may cause problems if a user is a member of two mutually exclusive groups. For example, a user who is a member of both the Movies and Electronics groups cannot see any data from the Product hierarchy, because that hierarchy does not contain any data that belongs to both the Movies and Electronics categories.

Prerequisites

To configure how security filters are merged, you must have the Configure Project Basic privilege.

To configure how Intelligence Server merges multiple security filters for a user or group

1. In Developer, log into a project. You must log in as a user with administrative privileges.
2. From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.
3. Expand the **Security Filter** category, and then select **General**.
4. Under **Security Filter Merge Options**, select one of the options:
   - **Union (OR) Security Filters on related attributes, intersect (AND) Security Filters on unrelated attributes** (see *Merging related security filters with OR and unrelated security filters with AND, page 87*)
   - **Intersect (AND) all Security Filters** (see *Merging all security filters with AND, page 88*)
5. Click **OK** to close the Project Configuration Editor.
6. Restart Intelligence Server for your changes to take effect.

 © 2016, MicroStrategy Inc.

Using a single security filter for multiple users: System prompts

A *system prompt* is a special type of prompt that does not require an answer from the user. Instead, it is answered automatically by Intelligence Server. System prompts are in the `Public Objects/Prompts/System Prompts` folder in Developer.

Note the following:

- Like other prompt objects, answers to system prompts are used to match caches. Therefore, users do not share caches for reports that contain different answers to system prompts.
- The system prompts Token 1, Token 2, Token 3, and Token 4 are provided to support using an XQuery source to authenticate users for a MicroStrategy project. For steps to report on and authenticate using XQuery sources, see the *Advanced Reporting Guide*.

The User Login prompt is a system prompt that is automatically answered with the login name of the user who executes the object containing the prompt. It can provide flexibility when implementing security mechanisms in MicroStrategy. You can use this prompt to insert the user's login name into any security filter, or any other object that can use a prompt.

If you are using LDAP authentication in your MicroStrategy system, you can import LDAP attributes into your system as system prompts. You can then use these system prompts in security filters, in the same way that you use the User Login system prompt, as described above. For instructions on how to import LDAP attributes as system prompts, see *Using LDAP attributes in security filters, page 131*.

For examples of how to use system prompts in security filters, see:

- *Simplifying the security filter definition process, page 90*
- *Implementing a report-level security filter, page 90*
- *Using database tables that contain security information, page 90*

To create a security filter using a system prompt

1. In Developer, from the **Administration** menu, point to **Projects** and then select **Security Filter Manager**. The Security Filter Manager opens.
2. From the **Choose a project** drop-down list, select the project that you want to create a security filter for.
3. Select the **Security Filters** tab.
4. Click **New**. The Security Filter Editor opens.
5. Double-click on the text **Double-click here to add a qualification**. The Filtering Options pane opens.
6. Select **Add an advanced qualification** and click **OK**.

© 2016, MicroStrategy Inc.

7 From the **Option** drop-down list, select **Custom Expression**.

8 Type your custom expression in the **Custom Expression** area. You can drag and drop a system prompt or other object to include it in the custom expression. For detailed instructions on creating custom expressions in filters, see the *Filters* chapter of the Advanced Reporting Guide.

9 When you have finished typing your custom expression, click **Validate** to make sure that its syntax is correct.

10 Click **Save and close**. Type a name for the security filter and click **Save**. The new security filter is saved.

Simplifying the security filter definition process

You can use a system prompt to apply a single security filter to all users in a group. For example, you can create a security filter using the formula `User@ID=?[User Login]` that displays information only for the element of the User attribute that matches the user's login.

For a more complex example, you can restrict Managers so that they can only view data on the employees that they supervise. Add the User Login prompt to a security filter in the form `Manager=?[User Login]`. Then assign the security filter to the Managers group. When a manager named John Smith executes a report, the security filter generates SQL for the condition `Manager='John Smith'` and only John Smith's employees' data is returned.

Implementing a report-level security filter

You can also use the User Login system prompt to implement security filter functionality at the report level, by defining a report filter with a system prompt. For example, you can define a report filter with the User Login prompt in the form `Manager=?[User Login]`. Any reports that use this filter return data only to those users who are listed as Managers in the system.

Using database tables that contain security information

If your organization maintains security information in database tables, you can use a system prompt to build MicroStrategy security mechanisms using the database security tables. For example, you can restrict the data returned based on a user's login by creating a report filter that accesses columns in your security tables and includes the User Login system prompt. You can also restrict data access based on two or more unrelated attributes by using logical views (database views) and the User Login system prompt in a security filter. For more information, including detailed instructions, on how to implement these examples, see MicroStrategy Tech Note TN11351.

 © 2016, MicroStrategy Inc.

Controlling access to data at the database (RDBMS) level

Database servers have their own security architectures that provide authentication, access control, and auditing. As mentioned above, you may choose to use these RDBMS techniques to manage access to data, or you may choose to use mechanisms in the MicroStrategy application layer to manage access to data, or you may use a combination of the two. They are not mutually exclusive. One advantage of using the database-level security mechanisms to secure data is that all applications accessing the database benefit from those security measures. If only MicroStrategy mechanisms are used, then only those users accessing the MicroStrategy application benefit from those security measures. If other applications access the database without going through the MicroStrategy system, the security mechanisms are not in place.

In addition to the security features that different database management systems offer, MicroStrategy is designed to work with general database security techniques such as:

- *Security views, page 91*
- *Splitting fact tables by rows, page 91*
- *Splitting fact tables by columns, page 93*

Security views

Most databases provide a way to restrict access to data. For example, a user may be able to access only certain tables, or he may be restricted to certain rows and columns within a table. The subset of data available to a user is called the user's security view.

Security views are often used when splitting fact tables by columns and splitting fact tables by rows (discussed below) cannot be used. The rules that determine which rows each user is allowed to see typically vary so much that users cannot be separated into a manageable number of groups. In the extreme, each user is allowed to see a different set of rows.

Note that restrictions on tables, or rows and columns within tables, may not be directly evident to a user. However, they do affect the values displayed in a report. You need to inform users as to which data they can access so that they do not inadvertently run a report that yields misleading final results. For example, if a user has access to only half of the sales information in the data warehouse but runs a summary report on all sales, the summary reflects only half of the sales. Reports do not indicate the database security view used to generate the report.

Consult your database vendor's product documentation to learn how to create security views for your database.

Splitting fact tables by rows

You can split fact tables by rows to separate a logical data set into multiple physical tables based on values in the rows (this is also known as table partitioning). The resultant tables are physically distinct tables in the data warehouse, and security administration is simple because permissions are granted to entire tables rather than to rows and columns.

If the data to be secured can be separated by rows, then this may be a useful technique. For example, suppose a fact table contains the key Customer ID, Address, Member Bank and two fact columns, as shown below:

Customer ID	Customer Address	Member Bank	Transaction Amount ($)	Current Balance ($)
123456	12 Elm St.	1st National	400.80	40,450.00
945940	888 Oak St.	Eastern Credit	150.00	60,010.70
908974	45 Crest Dr.	People's Bank	3,000.00	100,009.00
886580	907 Grove Rd.	1st National	76.35	10,333.45
562055	1 Ocean Blvd.	Eastern Credit	888.50	1,000.00

You can split the table into separate tables (based on the value in Member Bank), one for each bank: 1st National, Eastern Credit, and so on. In this example, the table for 1st National bank would look like this:

Customer ID	Customer Address	Member Bank	Transaction Amount ($)	Current Balance ($)
123456	12 Elm St.	1st National	400.80	40,450.00
886580	907 Grove Rd.	1st National	76.35	10,333.45

The table for Eastern Credit would look like this:

Customer ID	Customer Address	Member Bank	Transaction Amount ($)	Current Balance ($)
945940	888 Oak St.	Eastern Credit	150.00	60,010.70
562055	1 Ocean Blvd.	Eastern Credit	888.50	1,000.00

This makes it simple to grant permissions by table to managers or account executives who should only be looking at customers for a certain bank.

In most RDBMSs, split fact tables by rows are invisible to system users. Although there are many physical tables, the system "sees" one logical fact table.

Support for Split fact tables by rows for security reasons should not be confused with the support that Intelligence Server provides for split fact tables by rows for performance benefits. For more information on partitioning, see the *MicroStrategy Advanced Reporting Guide*.

 © 2016, MicroStrategy Inc.

Splitting fact tables by columns

You can split fact tables by columns to separate a logical data set into multiple physical tables by columns. If the data to be secured can be separated by columns, then this may be a useful technique.

Each new table has the same primary key, but contains only a subset of the fact columns in the original fact table. Splitting fact tables by columns allows fact columns to be grouped based on user community. This makes security administration simple because permissions are granted to entire tables rather than to columns. For example, suppose a fact table contains the key labeled Customer ID and fact columns as follows:

Customer ID	Customer Address	Member Bank	Transaction Amount ($)	Current Balance ($)

You can split the table into two tables, one for the marketing department and one for the finance department. The marketing fact table would contain everything except the financial fact columns as follows:

Customer ID	Customer Address	Member Bank

The second table used by the financial department would contain only the financial fact columns but not the marketing-related information as follows:

Customer ID	Transaction Amount ($)	Current Balance ($)

Merging users or groups

Within a given project source, you may need to combine multiple users into one user definition or combine a user group into another user group. For example, if UserA is taking over the duties of UserB, you may want to combine the users by merging UserB's properties into UserA. The MicroStrategy User Merge Wizard merges multiple users or groups and their profiles into a single user or group, with a single profile.

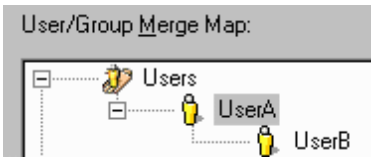
Topics covered in this section include:

- *How users and groups are merged, page 94*
- *Running the User Merge Wizard, page 96*

How users and groups are merged

The User Merge Wizard combines users and their related objects, from a single project source. These objects include profile folders, group memberships, user privileges, security roles, and security filters, among others. Information from the user or group that is being merged is copied to the destination user or group. Then the user or group that is being merged is removed from the metadata and only the destination user or group remains.

For example, you want to merge UserB into UserA. In this case UserA is referred to as the destination user. In the wizard, this is shown in the image below:



When you open the User Merge Wizard and select a project source, the wizard locks that project configuration. Other users cannot change any configuration objects until you close the wizard. For more information about locking and unlocking projects, see *Locking projects, page 231*.

You can also merge users in batches if you have a large number of users to merge. Merging in batches can significantly speed up the merge process. Batch-merging is an option in the User Merge Wizard. Click **Help** for details on setting this option.

The User Merge Wizard automatically merges the following properties: privileges, group memberships, profile folders, and object ownership (access control lists). You may optionally choose to merge properties such as a user's or group's security roles, security filters, and database connection maps. Details about how the wizard merges each of these properties are discussed below.

Merging user privileges

The User Merge Wizard automatically merges all of a user's or group's privileges. To continue with the example above, before the users are merged, each user has a distinct set of global user privileges. After the merge, all privileges that had been assigned to UserB are combined with those of the destination user, UserA. This combination is performed as a union. That is, privileges are not removed from either user.

For example, if UserA has the Web user privilege and UserB has the Web user and Web Administration privileges, after the merge, UserA has both Web user and Web Administration privileges.

 © 2016, MicroStrategy Inc.

Merging user group memberships

The User Merge Wizard automatically merges all of a user's or group's group memberships. Before the merge, each user has a distinct set of group memberships. After the merge, all group memberships that were assigned to UserB are combined with those of the destination user, UserA. This combination is performed as a union. That is, group memberships are not removed for either user.

Merging user profile folders

The User Merge Wizard automatically merges all of a user's or group's profile folders. Before the merge, UserA and UserB have separate and distinct user profile folders. After UserB is merged into UserA, only UserA exists; her profile contains the profile folder information from both UserA and UserB.

Merging object ownership and access control lists

The User Merge Wizard automatically merges all of a user's or group's object ownerships and access control lists (ACLs). Before the merge, the user to be merged, UserB, owns the user objects in her profile folder and also has full control over the objects in the access control list. After the merge, ownership and access to the merged user's objects are granted to the destination user, UserA. The merged user is removed from the object's ACL. Any other users that existed in the ACL remain in the ACL. For example, before the merge, UserB owns an object that a third user, UserC has access to. After the merge, UserA owns the object, and UserC still has access to it.

Merging project security roles

The User Merge Wizard does not automatically merge a user's or group's security roles. To merge them, you must select the **Security Roles** check box on the Merge Options page in the wizard. Before the merge, both users have unique security roles for a given project. After the merge, the destination user profile is changed based on the following rules:

- If neither user has a security role for a project, the destination user does not have a security role on that project.
- If the destination user has no security role for a project, the user inherits the role from the user to be merged.
- If the destination user and the user to be merged have different security roles, then the existing security role of the destination user is kept.
- If you are merging multiple users into a single destination user and each of the users to be merged has a security role, then the destination user takes the security role of the first user to be merged. If the destination user also has a security role, the existing security role of the destination user is kept.

© 2016, MicroStrategy Inc.

Merging project security filters

The User Merge Wizard does not automatically merge a user's or group's security filters. To merge them, you must select the **Security Filters** check box on the Merge Options page in the wizard. When merging security filters, the wizard follows the same rules as for security roles, described above.

Merging database connection mapping

The User Merge Wizard does not automatically merge a user's or group's database connection maps. To merge them, you must select the **Connection Mapping** check box on the Merge Options page in the wizard. When merging database connection mappings, the wizard follows the same rules as for security roles and security filters, described above.

Running the User Merge Wizard

The following high-level procedure provides an overview of what the User Merge Wizard does. For an explanation of the information required at any given page in the wizard, click **Help**, or press **F1**.

To merge users or groups

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **User Merge Wizard**. The User Merge Wizard opens.
2. Specify the project source containing the users/groups you wish to merge.
3. Select whether you wish to merge optional user properties such as security roles, security filters, and database connection maps. For a description of how the User Merge Wizard merges these optional properties, see each individual property's section in *How users and groups are merged, page 94*.
4. Specify whether you want to have the wizard select the users/groups to merge automatically (you can verify and correct the merge candidates), or if you wish to manually select them.
5. In the User Merge Candidates page, select the destination users or groups and click **>** to move them to the right-hand side.
6. Select the users or groups to be merged and click **>** to move them to the right-hand side. They display below the selected destination user or group.
7. On the Summary page, review your selections, and click **Finish**. The users or groups are merged.

 © 2016, MicroStrategy Inc.

3

Identifying Users: Authentication

Introduction

Authentication is the process by which the system identifies the user. In most cases, a user provides a login ID and password which the system compares to a list of authorized logins and passwords. If they match, the user is able to access certain aspects of the system, according to the access rights and application privileges associated with the user.

Workflow: changing authentication modes

The following is a list of high-level tasks that you perform when you change the default authentication mode in your MicroStrategy installation.

- Choose an authentication mode, and set up the infrastructure necessary to support it. For example, if you want to use LDAP Authentication, you must set up your LDAP directory and server. For the modes of authentication available, see *Modes of authentication, page 98*.
- Import your user database into the MicroStrategy metadata, or link your users' accounts in your user database with their accounts in MicroStrategy. For example, you can import users in your LDAP directory into the MicroStrategy metadata, and ensure that their LDAP credentials are linked to the corresponding MicroStrategy users. Depending on the authentication mode you choose, the following options are available:
    - If your organization's users do not exist in the MicroStrategy metadata:
        - You can import their accounts from an LDAP directory, or from a text file. For the steps to import users, refer to the *System Administration Help* in Developer.

© 2016, MicroStrategy Inc.

    - You can configure Intelligence Server to automatically import users into the metadata when they log in.
  - If your organization's users already exist in the MicroStrategy metadata:
    - You can use a Command Manager script to edit the user information in the metadata, and link the users' MicroStrategy accounts to their accounts in your user directory.
- Enable your chosen authentication mode for MicroStrategy applications at the following levels:
  - Your web server, for example, IIS or Apache.
  - Your application server, for example, IIS or WebSphere.
  - In Web Administrator, on the Default Server Properties page.
  - In Mobile Administrator, on the Default Server Properties page.
  - For all project sources that the above applications connect to.

  The specific steps to implement an authentication mode depend on the mode you choose, and are described in the sections that follow.

Modes of authentication

Several authentication modes are supported in the MicroStrategy environment. The main difference between the modes is the authentication authority used by each mode. The authentication authority is the system that verifies and accepts the login/password credentials provided by the user.

The available authentication modes are:

- **Standard**: Intelligence Server is the authentication authority. This is the default authentication mode. For more information, see *Implementing standard authentication, page 100*.
- **Anonymous**: Users log in as "Guest" and do not need to provide a password. This authentication mode may be required to enable other authentication modes, such as database warehouse or LDAP. For more information, see *Implementing anonymous authentication, page 102*.
- **Database warehouse**: The data warehouse database is the authentication authority. For more information, see *Implementing database warehouse authentication, page 103*.
- **LDAP (lightweight directory access protocol)**: An LDAP server is the authentication authority. For more information, see *Implementing LDAP authentication, page 105*.
- **Single sign-on**: Single sign-on encompasses several different third-party authentication methods, including:

© 2016, MicroStrategy Inc.

    - **Windows authentication**: Windows is the authentication authority. For more information, see *Implementing Windows NT authentication, page 133*.
    - **Integrated authentication**: A domain controller using Kerberos authentication is the authentication authority. For more information, see *Enabling integrated authentication, page 140*.
    - **Third-party authentication**: A third-party single sign-on tool, such as IBM® Tivoli® Access Manager, CA SiteMinder®, or Oracle® Access Manager, is the authentication authority. For more information, see *Enabling Single Sign-on to Web, Mobile, and Office with third-party authentication, page 161*.
- **MicroStrategy Usher**: Users log into Web and Mobile using MicroStrategy Usher. Usher enables users to electronically validate their identity using the Usher app and mobile badge on their smartphone, instead of entering a password. For steps, see *Enabling Usher authentication for Web and Mobile, page 177*.

For examples of situations where you might want to implement specific authentication modes, and the steps to do so, see *Authentication examples, page 180*.

Configuring the authentication mode for a project source

You can configure a project source to use a specific authentication mode using the Project Source Manager. By default, project sources use standard authentication (see *Implementing standard authentication, page 100*).

To configure the authentication mode for a project source

1. In Developer, from the **Tools** menu, select **Project Source Manager**. The Project Source Manager opens.
2. Select the appropriate project source and click **Modify**. The Project Source Manager for that project source opens.
3. On the **Advanced** tab, select the appropriate option for the default authentication mode that you want to use.
4. Click **OK** twice. The Project Source Manager closes and the specified authentication mode is now the default for that project source.
5. If the project source is accessed via MicroStrategy Web or MicroStrategy Office, there are additional steps that must be followed to configure the authentication mode, as follows:
    - To set the authentication mode in MicroStrategy Web, use the MicroStrategy Web Administrator's Default Server Properties page. For detailed instructions, see the *MicroStrategy Web Help*. (Click **Help** from the MicroStrategy Web Administrator page.)

© 2016, MicroStrategy Inc.

- To set the authentication mode in MicroStrategy Office, use the `projectsources.xml` file. For detailed instructions, see the MicroStrategy Office User Guide.

Importing users from different authentication systems

You can import users from multiple different authentication systems, such as from a database warehouse and from an LDAP Server, into a single MicroStrategy metadata.

Each user that is imported into MicroStrategy from a single authentication mechanism is created as a separate user object in the MicroStrategy metadata. For example, if User A is imported from your LDAP Server into MicroStrategy, the User A object is created in the MicroStrategy metadata. If User A is also imported from your NT system, a separate User A object (we can call it User A-NT) is created in the metadata. Every time a user is imported into the MicroStrategy metadata, a separate user object is created.

As an alternative, you can import User A from a single authentication system (LDAP, for example), and then link the User A object that is created to the same user in your NT system, and to the same user in your database warehouse, and so on. Using linking, you can "connect" or map multiple authentication systems to a single user object in the MicroStrategy metadata.

Sharing user accounts between users

MicroStrategy does not recommend sharing user accounts.

You may decide to map several users to the same MicroStrategy user account. These users would essentially share a common login to the system. Consider doing this only if you have users who do not need to create their own individual objects, and if you do not need to monitor and identify each individual user uniquely.

Implementing standard authentication

Standard authentication is the default authentication mode and the simplest to set up. Each user has a unique login and password and can be identified in the MicroStrategy application uniquely.

By default, all users connect to the data warehouse using one RDBMS login ID, although you can change this using Connection Mapping. For more information, see *Connecting to the data warehouse, page 8*. In addition, standard authentication is the only authentication mode that allows a user or system administrator to change or expire MicroStrategy passwords.

 © 2016, MicroStrategy Inc.

When using standard authentication, Intelligence Server is the authentication authority. Intelligence Server verifies and accepts the login and password provided by the user. This information is stored in the metadata repository.

When a project source is configured to use standard authentication, users must enter a valid login ID and password combination before they can access the project source.

Password policy

A valid password is a password that conforms to any specifications you may have set. You can define the following characteristics of passwords:

- Whether a user must change his password when he first logs into MicroStrategy
- How often the password expires
- The number of past passwords that the system remembers, so that users cannot use the same password
- Rules for password complexity, including:
  - The minimum number of characters that the password must contain
  - The minimum number of upper-case characters that the password must contain
  - The minimum number of lower-case characters that the password must contain
  - The minimum number of numeric characters, that is, numbers from 0 to 9, that the password must contain
  - The minimum number of special characters, that is, symbols, that the password must contain

The expiration settings are made in the User Editor and can be set for each individual user. The complexity and remembered password settings are made in the Security Policy Settings dialog box, and affect all users. For detailed information about configuring these settings, see the *MicroStrategy Developer Help*.

Steps to implement standard authentication

The procedure below gives the high-level steps for configuring your Intelligence Server for standard authentication. For additional information about any of these steps, see the *MicroStrategy Developer Help*.

High-level steps to configuration standard authentication

1. In Developer, open the Project Source Manager and click **Modify**.
2. On the Advanced tab, select **Use login ID and password entered by the user (standard authentication)**. (This is the default setting.)

© 2016, MicroStrategy Inc.

3 In MicroStrategy Web, log in as an administrator. On the **Preferences** page, select **Project Defaults**, select **Security**, and then enable **Standard (user name & password)** as the login mode.

4 In Developer, create a database instance for the data warehouse and assign it a default database login. This is the RDBMS account that will be used to execute reports from all users.

Implementing anonymous authentication

When using anonymous authentication, users log in as guests and do not need to provide a password. Each guest user assumes the profile defined by the Public group. When a user logs in as a guest, a new user is created dynamically and becomes a member of the Public group.

This dynamically created guest user is not the same as the “Guest” user which is visible in the User Manager.

Guest users inherit security settings, including privileges and permissions, project access, security filter, and connection map information, from the Public/Guest group; they are not part of the Everyone group.

By default, guest users have no privileges; you must assign this group any privileges that you want the guest users to have. Privileges that are grayed out in the User Editor are not available by default to a guest user. Other than the unavailable privileges, you can determine what the guest user can and cannot do by modifying the privileges of the Public/Guest user group and by granting or denying it access to objects. For more information, see *Controlling access to functionality: Privileges, page 66* and *Controlling access to objects: Permissions, page 58*.

All objects created by guest users must be saved to public folders and are available to all guest users. Guest users may use the History List, but their messages in the History List are not saved and are purged when the guest users log out.

To enable anonymous access to a project source

By default, anonymous access is disabled at both the server and the project levels.

1 In Developer, log into the project source with a user that has administrative privileges.

2 From the folder List, select **Administration**.

3 From the **File** menu, select **Properties**. The Properties - *project source name* dialog box opens.

 © 2016, MicroStrategy Inc.

4 In the **Security** tab, click **Add**. The Select Developer Users and Groups dialog box opens.

5 Select the Public/Guest group.

6 In the **Access Permission** list, select **Connect**.

7 Click **OK**. The Select Developer Users and Groups dialog box closes.

8 Click **OK**. The Properties dialog box closes.

9 Follow the procedure in *Configuring the authentication mode for a project source, page 99* and select **Anonymous authentication**. When users log into this project source, they are now automatically logged in as guest users and not prompted for a login or password.

Implementing database warehouse authentication

This mode of authentication identifies users by means of a login ID and password stored in the data warehouse database. The RDBMS is the authentication authority and verifies that the login ID and password are valid. Each report is executed on the RDBMS under the RDBMS account of the user who submitted the report from the MicroStrategy system. Users log in to the MicroStrategy system with their RDBMS login and password, and each MicroStrategy user is linked to an RDBMS account.

Use database warehouse authentication if you want the data warehouse RDBMS to be the authority for identifying users and you do not want to maintain user credentials in Intelligence Server as well as the RDBMS. You can also use this configuration if you need to keep an audit trail at the RDBMS level for each query executed by each individual user.

If you use database authentication, for security reasons MicroStrategy recommends that you use the setting **Create caches per database login**. This ensures that users who execute their reports using different database login IDs cannot use the same cache. You can set this in the Project Configuration Editor in the **Caching**: **Result Caches**: **Creation** category.

Database warehouse authentication information flow

The following scenario presents an overview of the general flow of information between Intelligence Server and a database server when a database user logs into Developer or MicroStrategy Web.

1 The user anonymously logs in to a project source.

This is done anonymously because the user has not yet logged in to a specific project. Because a warehouse database is not associated with the project source itself, users are not authenticated until they select a project to use. For more information about anonymous authentication, including instructions on enabling it for a project source, see *Implementing anonymous authentication, page 102*.

Note the following:

- By default, the Public/Guest group is denied access to all projects. A security role with View access to the projects must be explicitly assigned to the Public/Guest group, so that these users can see and log in to the available projects.
- All users logging in to a database warehouse authentication project source can see all projects visible to the Guest user. Project access is then granted or denied for each individual user when the user attempts to log into the project.

2. The user selects a project, and then logs in to that project using her data warehouse login ID and password. She is authenticated against the data warehouse database associated with that project.

   To enable database authentication, you must link the users in the MicroStrategy metadata to RDBMS users. Linking causes Intelligence Server to map a warehouse database user to a MicroStrategy user. If a user attempts to log in to a project without having been linked to a MicroStrategy user, a "User not found" error message is returned.

Steps to implement database warehouse authentication

The procedure below gives the high-level steps for configuring your Intelligence Server for database warehouse authentication. For additional information about any of these steps, see the *MicroStrategy Developer Help*.

High-level steps for configuring database warehouse authentication

1. Create a DSN and a database instance for the authentication database.
2. Configure the project source to allow anonymous authentication (see *Implementing standard authentication, page 100*).
3. Configure the project source to use database warehouse authentication (see *Configuring the authentication mode for a project source, page 99*).
4. Assign a security role to the Public/Guest group for each project to which you want to provide access (see *Managing security roles, page 70*).

 © 2016, MicroStrategy Inc.

5 Link each MicroStrategy user to an RDBMS user. In the User Editor, in the **Authentication**: **Metadata** category, type the data warehouse login ID in the **Database Login** field.

You can create the MicroStrategy users by importing a list of the RDBMS users into the MicroStrategy system. For instructions, see *Creating, importing, and deleting users and groups, page 56*.

6 For each project, in the Project Configuration Editor, in the **Database instances**: **Authentication**: **Metadata** category, specify the database instance for the authentication database.

7 For each project, enable database execution using linked warehouse logins (see *Linking database users and MicroStrategy users: Passthrough execution, page 77*).

8 To enable database authentication in MicroStrategy Web, log in as an administrator. On the **Preferences** page, select **Project Defaults**. The Project Defaults page is displayed.

9 Under Security, select the **Database Authentication** check box, and then click **Apply**.

Database warehouse and metadata (6.x) authentication

This mode of authentication uses the login ID and password entered by the user for both the metadata and data warehouse databases. It is available only for connecting to MicroStrategy 6.X project sources.

Under this authentication mode, Intelligence Server passes the login credentials to the metadata database when you log in to the project source. You implement this authentication mode by following the same general steps as for database warehouse authentication, in *Steps to implement database warehouse authentication, page 104*.

Implementing LDAP authentication

Lightweight Directory Access Protocol (LDAP) is an open standard Internet protocol running over TCP/IP that is designed to maintain and work with large user directory services. It provides a standard way for applications to request and manage user and group directory information. LDAP performs simple `Select` operations against large directories, in which the goal is to retrieve a collection of attributes with simple qualifications, for example, `Select all the employees' phone numbers in the support division`.

An LDAP authentication system consists of two components: an LDAP server and an LDAP directory. An LDAP server is a program that implements the LDAP protocol and controls access to an LDAP directory of user and group accounts. An LDAP directory is

the storage location and structure of user and group accounts on an LDAP server. Before information from an LDAP directory can be searched and retrieved, a connection to the LDAP server must be established.

If you use an LDAP directory to centrally manage users in your environment, you can implement LDAP authentication in MicroStrategy. Group membership can be maintained in the LDAP directory without having to also be defined in Intelligence Server. LDAP authentication identifies users in an LDAP directory which MicroStrategy can connect to through an LDAP server. Supported LDAP servers include Novell Directory Services, Microsoft Directory Services, and Sun ONE 5.1/iPlanet. For the latest set of certified and supported LDAP servers, refer to the *MicroStrategy Readme*.

The high-level steps to implement LDAP authentication are as follows:

1. Review the LDAP information flow, described in *LDAP information flow, page 106*.
2. Depending on your requirements, collect information and make decisions regarding the information in *Checklist: Information required for connecting your LDAP server to MicroStrategy, page 107*.
3. Run the LDAP Connectivity Wizard to connect your LDAP server to MicroStrategy, as described in *Setting up LDAP authentication in MicroStrategy, page 123*.
4. To make changes in your LDAP configuration, use the procedures described in *Managing LDAP authentication, page 125*.

You can also set up MicroStrategy Office to use LDAP authentication. For information, see the MicroStrategy Office User Guide.

LDAP information flow

The following scenario presents a high-level overview of the general flow of information between Intelligence Server and an LDAP server when an LDAP user logs into Developer or MicroStrategy Web:

1. When an LDAP user logs in to MicroStrategy Web or Developer, Intelligence Server connects to the LDAP server using the credentials for the LDAP administrative user, called an authentication user.
2. The authentication user is bound to LDAP using a Distinguished Name (DN) and password set up in the user's configuration.
3. The authentication user searches the LDAP directory for the user who is logging in via Developer or MicroStrategy Web, based on the DN of the user logging in.
4. If this search successfully locates the user who is logging in, the user's LDAP group information is retrieved.
5. Intelligence Server then searches the MicroStrategy metadata to determine whether the DN of the user logging in is linked to an existing MicroStrategy user or not.
6. If a linked user is not found in the metadata, Intelligence Server refers to the import and synchronization options that are configured. If importing is enabled, Intelligence Server updates the metadata with the user and group information it accessed in the LDAP directory.

 © 2016, MicroStrategy Inc.

7 If a linked user is not found and importing is disabled, but the LDAP server is configured to accept anonymous authentication, Intelligence Server creates a new user session with an anonymous Guest user. If the LDAP server is not configured to accept anonymous authentication, Intelligence Server does not allow the user to log in.

8 The user who is logging in is given access to MicroStrategy, with appropriate privileges and permissions.

Checklist: Information required for connecting your LDAP server to MicroStrategy

You can connect your LDAP server to your Intelligence Server using the LDAP Connectivity Wizard. Before beginning the process, ensure that you have the following information:

- The connection details for your LDAP server. The information required is as follows:
  - The machine name or IP address of the LDAP server.
  - The network port that the LDAP server uses.
  - Whether the LDAP server is accessed using clear text, or over an encrypted SSL connection. If you are using an SSL connection, you need to do the following before you begin to set up LDAP:
    a Obtain a valid certificate from your LDAP server and save it on the machine where Intelligence Server is installed.
    b Follow the procedure recommended by your operating system to install the certificate.
  - The user name and password of an LDAP user who can search the LDAP directory. This user is called the authentication user, and is used by the Intelligence Server to connect to the LDAP server. Typically, this user has administrative privileges for your LDAP server.
- Details of your LDAP SDK. The LDAP SDK is a set of connectivity file libraries (DLLs) that MicroStrategy uses to communicate with the LDAP server. For information on the requirements for your LDAP SDK, and for steps to set up the SDK, see *Setting up LDAP SDK connectivity, page 110*.
- Your LDAP search settings, which allow Intelligence Server to effectively search through your LDAP directory to authenticate and import users. For information on defining LDAP search settings, see *Defining LDAP search filters to verify and import users and groups at login, page 112*.

Additionally, depending on your organization's requirements, it is recommended that you make decisions and gather information about the following:

- Determine whether you want to use connection pooling with your LDAP server. With connection pooling, you can reuse an open connection to the LDAP server for subsequent operations. The connection to the LDAP server remains open even when the connection is not processing any operations (also known as pooling). This setting

© 2016, MicroStrategy Inc.

can improve performance by removing the processing time required to open and close a connection to the LDAP server for each operation.

For background information on connection pooling, see *Determining whether to use connection pooling, page 115*.

- Determine the method that Intelligence Server uses to authenticate users in the LDAP server. The possible options are described below:
  - Binding: If you choose this method, the Intelligence Server attempts to log in to the LDAP server with the user's credentials. Intelligence Server also checks the LDAP server for user's restrictions, such as whether the user's account has been locked, or the user's password has expired.
  - Password comparison: If you choose this method, the Intelligence Server verifies the user's user name and password with the LDAP server, without attempting to log in to the LDAP server.

  For a comparison of the two methods of authentication, see *Determining whether to use authentication binding or password comparison, page 117*.
- Determine whether you need to use database passthrough authentication. In MicroStrategy, a single user name and password combination is frequently used to connect to and execute jobs against a database. However, you can choose to pass to the database a user's LDAP user name and password used to log in to MicroStrategy. The database is then accessed and jobs are executed using the LDAP user name and password. This allows each user logged in to MicroStrategy to execute jobs against the database using their unique user name and password which can be given a different set of privileges than other users.

  For additional information on database passthrough authentication, see *Determining whether to enable database passthrough authentication with LDAP, page 117*.
- Determine whether you want to import LDAP user and group information into the MicroStrategy metadata. The following options are available:
  - Import users and groups into MicroStrategy: If you choose this option, a MicroStrategy user is created for each user in your LDAP directory. Users can then be assigned additional privileges and permissions in MicroStrategy.
  - Link users and groups to MicroStrategy, without importing them: If you choose this option, a link is created between MicroStrategy users and users in your LDAP directory, without creating new LDAP users in your metadata. If you have an LDAP directory with a large number of users, this option avoids filling your metadata with new users.

  For information on the benefits and considerations for importing LDAP user and group information into MicroStrategy, see *Determining whether to import LDAP users into MicroStrategy, page 118*.
- Determine whether you want to automatically synchronize user and group information with the LDAP server. This ensures that if there are changes in the group membership for the users you have imported into MicroStrategy, or users who are linked to existing MicroStrategy accounts, the changes in the LDAP directory are applied in MicroStrategy when users log in, or on a schedule that you determine.

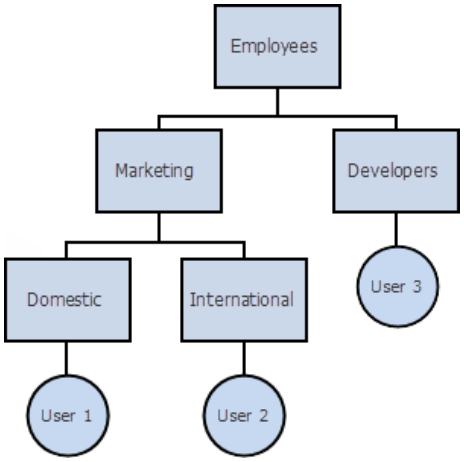
© 2016, MicroStrategy Inc.

For the benefits and considerations of synchronizing user and group information, see *Determining whether to automatically synchronize LDAP user and group information, page 121*.

- If you choose to import LDAP user and group information into the MicroStrategy metadata, determine the following:
  - Determine whether you want to import LDAP user and group information into the MicroStrategy metadata when users log in, and whether the information is synchronized every time users log in.
  - Determine whether you want to import LDAP user and group information into the MicroStrategy metadata in batches, and whether you want the information to be synchronized according to a schedule.

    If you want to import LDAP user and group information in batches, you must provide search filters to import the users. For example, if your organization has 1,000 users in the LDAP directory, of whom 150 need to use MicroStrategy, you must provide a search filter that imports the 150 users into the MicroStrategy metadata. For information on defining search filters, see *Defining LDAP search filters to verify and import users and groups at login, page 112*.
  - If your LDAP organizational structure includes groups contained within groups, determine how many recursive groups to import when you import a user or group into MicroStrategy.

    To understand how this setting effects the way the users and groups are imported into MicroStrategy, see the following diagram:



    If you choose to import two nested groups when MicroStrategy imports LDAP groups, the groups associated with each user are imported, up to two levels above the user. In this case, for User 1, the groups Domestic and Marketing would be imported. For User 3, Developers and Employees would be imported.
  - If you use a single sign-on (SSO) authentication system, such as Windows authentication or integrated authentication, determine whether you want to import the LDAP user and group information for users of your single sign-on system.
  - Determine whether the following additional information is imported:

© 2016, MicroStrategy Inc.

— The users' email addresses. If you have a license for MicroStrategy Distribution Services, then when you import LDAP users, you can import these email addresses as contacts associated with those users.

— The format that the users' login IDs and group names are imported in. For example, you can determine whether users log in to MicroStrategy using their LDAP login ID, or their LDAP distinguished name.

— Additional LDAP attributes to import. For example, your LDAP directory may include an attribute called `accountExpires`, which contains information about when the users' accounts expire. The attributes in your LDAP directory depend on the LDAP server that you use, and your LDAP configuration.

You can create security filters based on the LDAP attributes that you import. For example, you import the LDAP attribute `countryName`, create a security filter based on that LDAP attribute, and then you assign that security filter to all LDAP users. Now, when a user from Brazil views a report that breaks down sales revenue by country, she only sees the sales data for Brazil.

For information on setting up security filters based on LDAP attributes, see *Using LDAP attributes in security filters, page 131*.

Once you have collected the above information, you can use the LDAP Connectivity Wizard to set up your LDAP connection. The steps are described in *Setting up LDAP authentication in MicroStrategy, page 123*.

Setting up LDAP SDK connectivity

From the perspective of your LDAP server, Intelligence Server is an LDAP client that uses clear text or encrypted SSL to connect to your LDAP server through the LDAP SDK.

The LDAP SDK is a set of connectivity file libraries (DLLs) that MicroStrategy uses to communicate with the LDAP server. For the latest set of certified and supported LDAP SDK files, refer to the *MicroStrategy Readme*.

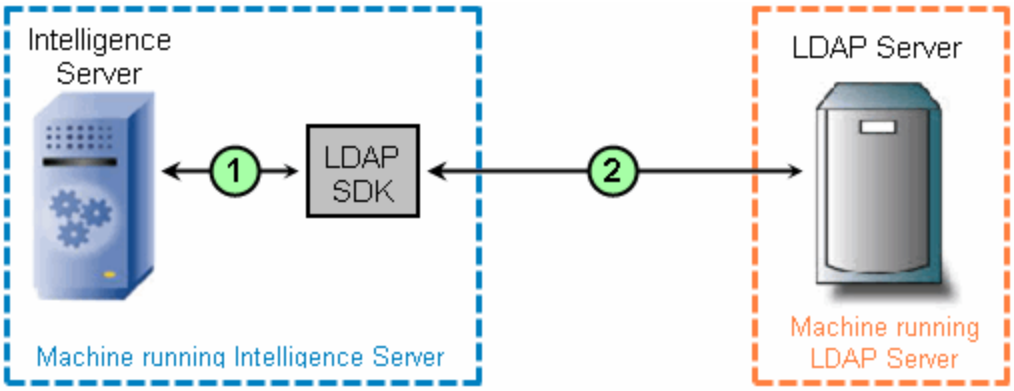
Intelligence Server requires that the version of the LDAP SDK you are using supports the following:

- LDAP v. 3
- SSL connections
- 64-bit architecture on UNIX and Linux platforms

In order for LDAP to work properly with Intelligence Server Universal, the 64-bit LDAP libraries must be used.

The following image shows how behavior of the various elements in an LDAP configuration affects other elements in the configuration.

© 2016, MicroStrategy Inc.



1 The behavior between Intelligence Server and the LDAP SDK varies slightly depending on the LDAP SDK used. The *MicroStrategy Readme* provides an overview of these behaviors.

2 The behavior between the LDAP SDK and the LDAP server is identical, no matter which LDAP SDK is used.

MicroStrategy recommends that you use the LDAP SDK vendor that corresponds to the operating system vendor on which Intelligence Server is running in your environment. Specific recommendations are listed in the *MicroStrategy Readme*, with the latest set of certified and supported LDAP SDKs, references to MicroStrategy Tech Notes with version-specific details, and SDK download location information.

To configure Intelligence Server to use specific DLLs, see the *Intelligence Server Configuration Editor: LDAP category, Platform* section in the *MicroStrategy Developer Help*.

High-level steps to install the LDAP SDK DLLs

1 Download the LDAP SDK DLLs onto the machine where Intelligence Server is installed.

2 Install the LDAP SDK.

3 Register the location of the LDAP SDK files as follows:

- Windows environment: Add the path of the LDAP SDK libraries as a system environment variable so that Intelligence Server can locate them.
- UNIX/Linux environment: Modify the `LDAP.sh` file located in the `env` folder of your MicroStrategy installation to point to the location of the LDAP SDK libraries. The detailed procedure is described in the procedure *To add the LDAP SDK path to the environment variable in UNIX, page 112* below.

4 Restart Intelligence Server.

© 2016, MicroStrategy Inc.

To add the LDAP SDK path to the environment variable in UNIX

This procedure assumes you have installed an LDAP SDK. For high-level steps to install an LDAP SDK, see *High-level steps to install the LDAP SDK DLLs, page 111*.

1. In a UNIX/Linux console window, browse to `<HOME_PATH>` where `<HOME_PATH>` is the directory you specified as the home directory during MicroStrategy installation. Browse to the folder `/env` in this path.
2. Add `Write` privileges to the `LDAP.sh` file by typing the command `chmod u+w LDAP.sh` and then pressing `ENTER`.
3. Open the `LDAP.sh` file in a text editor and add the library path to the `MSTR_LDAP_LIBRARY_PATH` environment variable. For example: `MSTR_LDAP_LIBRARY_PATH='/path/LDAP/library'`

   It is recommended that you store all libraries in the same path. If you have several paths, you can add all paths to the `MSTR_LDAP_LIBRARY_PATH` environment variable and separate them by a colon (:). For example: `MSTR_LDAP_LIBRARY_PATH='/path/LDAP/library:/path/LDAP/library2'`
4. Remove Write privileges from the `LDAP.sh` file by typing the command `chmod a-w LDAP.sh` and then pressing `ENTER`.
5. Restart Intelligence Server for your changes to take effect.

Defining LDAP search filters to verify and import users and groups at login

You must provide Intelligence Server with some specific parameters so it can search effectively through your LDAP directory for user information.

When users attempt to log in to MicroStrategy, the Intelligence Server authenticates users by searching the LDAP directory for the user's Distinguished Name, which is a unique way to identify users within the LDAP directory structure.

To search effectively, Intelligence Server must know where to start its search. When setting up LDAP authentication, it is recommended that you indicate a search root Distinguished Name to establish the directory location from which Intelligence Server starts all user and group searches. If this search root is not set, Intelligence Server searches the entire LDAP directory.

Additionally, you can specify search filters, which help narrow down the users and groups to search.

The following sections describe the search settings that you can configure:

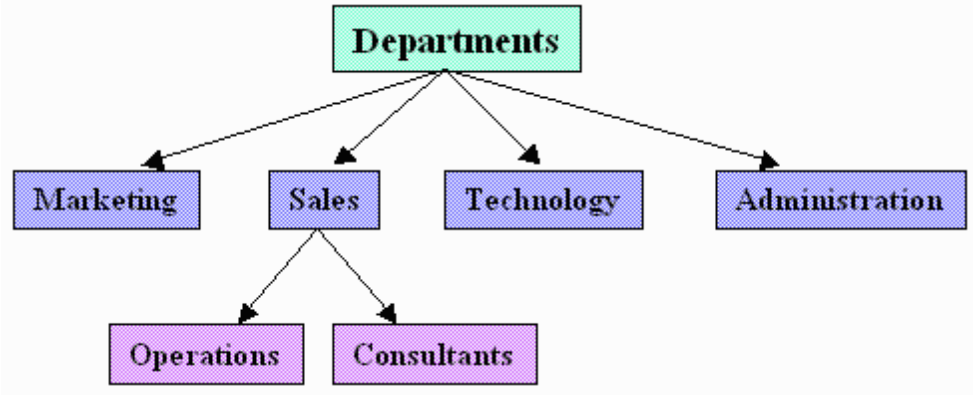
- *Highest level to start an LDAP search: Search root, page 113* provides examples of these parameters as well as additional details of each parameter and some LDAP server-specific notes.

© 2016, MicroStrategy Inc.

- *Finding users: user search filters, page 114* provides an overview of LDAP user search filters.
- *Finding groups: group search filters, page 114* provides an overview of LDAP group search filters.

Highest level to start an LDAP search: Search root

The following diagram and table present several examples of possible search roots based on how users might be organized within a company and within an LDAP directory. The diagram shows a typical company's departmental structure. The table describes several user import scenarios based on the diagram.

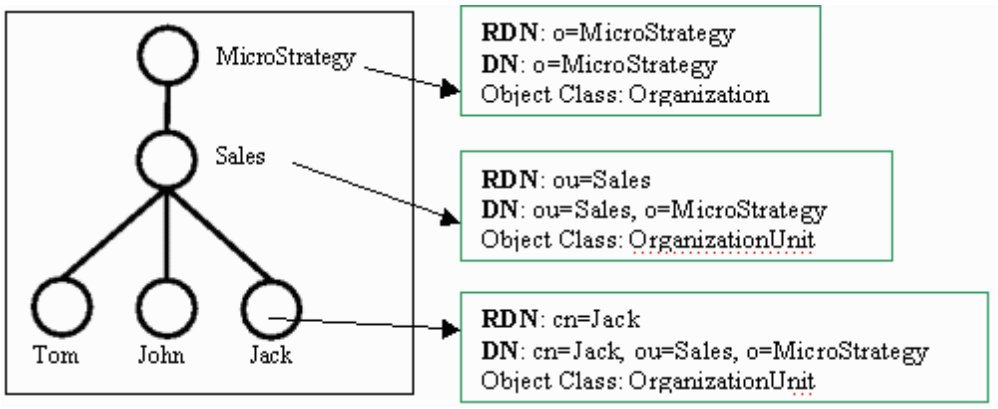


The following table, based on the diagram above, provides common search scenarios for users to be imported into MicroStrategy. The search root is the root to be defined in MicroStrategy for the LDAP directory.

Scenario	Search Root
Include all users and groups from Operations	Operations
Include all users and groups from Operations, Consultants, and Sales	Sales
Include all users and groups from Operations, Consultants, and Technology	Departments (with an exclusion clause in the User/Group search filter to exclude users who belong to Marketing and Administration)
Include all users and groups from Technology and Operations but not Consultants.	Departments (with an exclusion clause in the User/Group search filter to exclude users who belong to Consultants.)

For some LDAP vendors, the search root cannot be the LDAP tree's root. For example, both Microsoft Active Directory and Sun ONE require a search to begin from the domain controller RDN (dc). The image below shows an example of this type of RDN, where "dc=sales, dc=microstrategy, dc=com":

© 2016, MicroStrategy Inc.



If your LDAP directory has multiple domains for different departments, see MicroStrategy Tech Note TN18229.

Finding users: user search filters

User search filters allow MicroStrategy to efficiently search an LDAP directory to authenticate or import a user at login.

Once Intelligence Server locates the user in the LDAP directory, the search returns the user's Distinguished Name, and the password entered at user login is verified against the LDAP directory. Intelligence Server uses the authentication user to access, search in, and retrieve the information from the LDAP directory.

Using the user's Distinguished Name, Intelligence Server searches for the LDAP groups that the user is a member of. You must enter the group search filter parameters separately from the user search filter parameters (see *Finding groups: group search filters, page 114*).

User search filters are generally in the form `(&(objectclass=`*`LDAP_USER_OBJECT_CLASS`*`)(`*`LDAP_LOGIN_ATTR`*`=#LDAP_LOGIN#))` where:

- *`LDAP_USER_OBJECT_CLASS`* indicates the object class of the LDAP users. For example, you can enter `(&(objectclass=`**`person`**`)(cn=#LDAP_LOGIN#))`.
- *`LDAP_LOGIN_ATTR`* indicates which LDAP attribute to use to store LDAP logins. For example, you can enter `(&(objectclass=person)(`**`cn`**`=#LDAP_LOGIN#))`.
- `#LDAP_LOGIN#` can be used in this filter to represent the LDAP user login.

Depending on your LDAP server vendor and your LDAP tree structure, you may need to try different attributes within the search filter syntax above. For example, `(&(objectclass=person)(`*`uniqueID`*`=#LDAP_LOGIN#))`, where *`uniqueID`* is the LDAP attribute name your company uses for authentication.

Finding groups: group search filters

Group search filters allow MicroStrategy to efficiently search an LDAP directory for the groups to which a user belongs. These filters can be configured in the Intelligence Server

© 2016, MicroStrategy Inc.

Configuration Editor, under the LDAP subject.

The group search filter is generally in one of the following forms (or the following forms may be combined, using a pipe | symbol to separate the forms):

- `(&(objectclass=`*`LDAP_GROUP_OBJECT_CLASS`*`) (`*`LDAP_MEMBER_LOGIN_ATTR`*`=#LDAP_LOGIN#))`
- `(&(objectclass=`*`LDAP_GROUP_OBJECT_CLASS`*`) (`*`LDAP_MEMBER_DN_ATTR`*`=#LDAP_DN#))`
- `(&(objectclass=`*`LDAP_GROUP_OBJECT_CLASS`*`) (gidNumber=#LDAP_GIDNUMBER#))`

The group search filter forms listed above have the following placeholders:

- *`LDAP_GROUP_OBJECT_CLASS`* indicates the object class of the LDAP groups. For example, you can enter `(&(objectclass=`**`groupOfNames`**`)(member=#LDAP_DN#))`.
- *`LDAP_MEMBER_[LOGIN or DN]_ATTR`* indicates which LDAP attribute of an LDAP group is used to store LDAP logins/DNs of the LDAP users. For example, you can enter `(&(objectclass=groupOfNames)(`**`member`**`=#LDAP_DN#))`.
- `#LDAP_DN#` can be used in this filter to represent the distinguished name of an LDAP user.
- `#LDAP_LOGIN#` can be used in this filter to represent an LDAP user's login.
- `#LDAP_GIDNUMBER#` can be used in this filter to represent the UNIX or Linux group ID number; this corresponds to the LDAP attribute `gidNumber`.

You can implement specific search patterns by adding additional criteria. For example, you may have 20 different groups of users, of which only five groups will be accessing and working in MicroStrategy. You can add additional criteria to the group search filter to import only those five groups.

Determining whether to use connection pooling

With connection pooling, you can reuse an open connection to the LDAP server for subsequent operations. The connection to the LDAP server remains open even when the connection is not processing any operations (also known as pooling). This setting can improve performance by removing the processing time required to open and close a connection to the LDAP server for each operation.

If you do not use connection pooling, the connection to an LDAP server is closed after each request. If requests are sent to the LDAP server infrequently, this can help reduce the use of network resources.

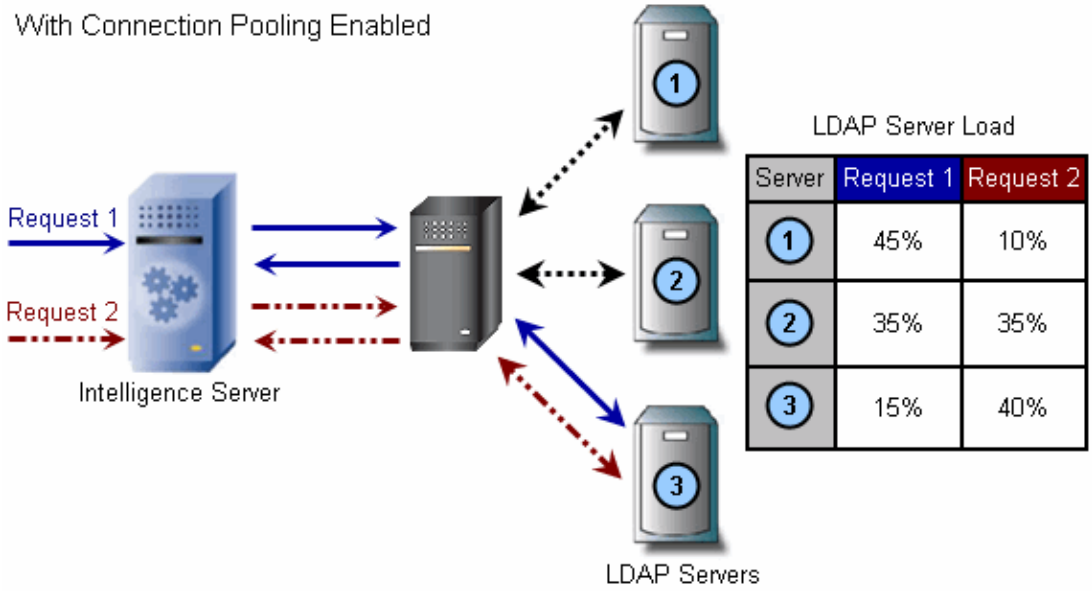
Connection pooling with clustered LDAP servers

You may have multiple LDAP servers which work together as a cluster of LDAP servers.

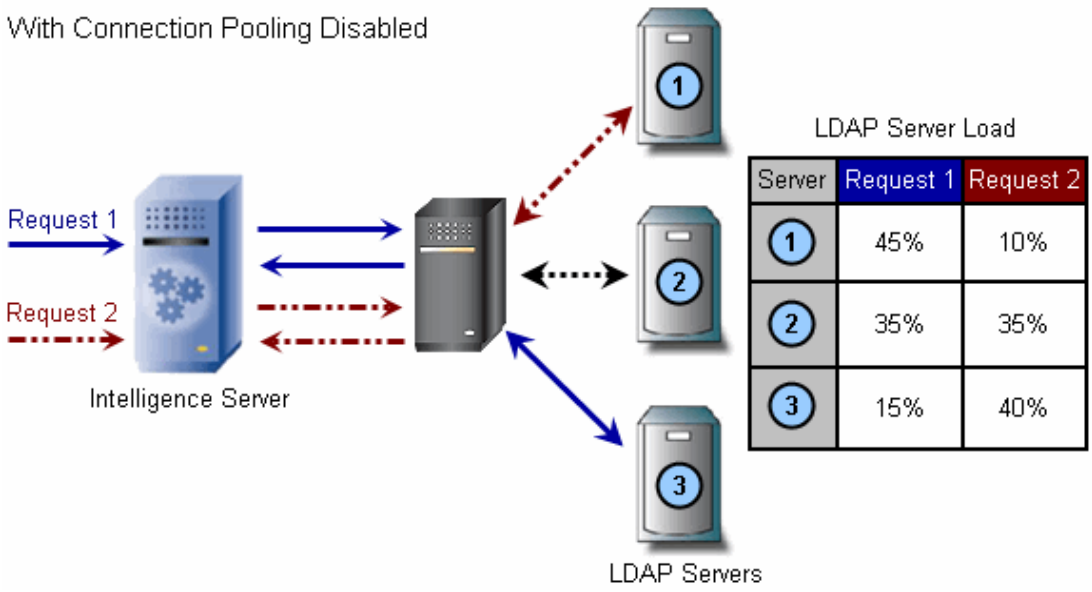
If connection pooling is disabled, when a request to open an LDAP connection is made, the LDAP server with the lightest load at the time of the request is accessed. The operation against the LDAP directory can then be completed, and in an environment without connection pooling, the connection to the LDAP server is closed. When the next request to open an LDAP connection is made, the LDAP server with the least amount of load is determined again and chosen.

If you enable connection pooling for a clustered LDAP environment, the behavior is different than described above. On the first request to open an LDAP connection, the LDAP server with the least amount of load at the time of the request is accessed. However, the connection to the LDAP server is not closed because connection pooling is enabled. Therefore, instead of determining the LDAP server with the least amount of load during the next request to open an LDAP connection, the currently open connection is reused.

The diagrams shown below illustrate how subsequent connections to a clustered LDAP server environment are handled, depending on whether connection pooling is enabled or disabled.



© 2016, MicroStrategy Inc.



Determining whether to use authentication binding or password comparison

When MicroStrategy attempts to authenticate an LDAP user logging in to MicroStrategy, you can choose to perform an LDAP bind to authenticate the user or simply authenticate on user name and password.

By implementing authentication binding, MicroStrategy authenticates the user by logging in to the LDAP server with the user's credentials, and assessing the following user restrictions:

- Whether the LDAP password is incorrect, has been locked out, or has expired
- Whether the LDAP user account has been disabled, or has been identified as an intruder and is locked out

If MicroStrategy can verify that none of these restrictions are in effect for this user account, MicroStrategy performs an LDAP bind, and successfully authenticates the user logging in. This is the default behavior for users and groups that have been imported into MicroStrategy.

You can choose to have MicroStrategy verify only the accuracy of the user's password with which the user logged in, and not check for additional restrictions on the password or user account. To support password comparison authentication, your LDAP server must also be configured to allow password comparison only.

Determining whether to enable database passthrough authentication with LDAP

In MicroStrategy, a single user name and password combination is frequently used to connect to and execute jobs against a database. However, you can choose to pass a user's LDAP user name and password used to log in to MicroStrategy to the database. The

© 2016, MicroStrategy Inc.

database is then accessed and jobs are executed using the LDAP user name and password. This allows each user logged in to MicroStrategy to execute jobs against the database using their unique user name and password, which can be given a different set of privileges than other users.

Database passthrough authentication is selected for each user individually. For general information on selecting user authentication, see *About MicroStrategy users, page 52*.

If a user's password is changed during a session in MicroStrategy, scheduled tasks may fail to run when using database passthrough authentication. Consider the following scenario.

A user with user login UserA and password PassA logs in to MicroStrategy at 9:00 A.M. and creates a new report. The user schedules the report to run at 3:00 P.M. later that day. Since there is no report cache, the report will be executed against the database. At noon, an administrator changes UserA's password to PassB. UserA does not log back into MicroStrategy, and at 3:00 P.M. the scheduled report is run with the credentials UserA and PassA, which are passed to the database. Since these credentials are now invalid, the scheduled report execution fails.

To prevent this problem, schedule password changes for a time when users are unlikely to run scheduled reports. In the case of users using database passthrough authentication who regularly run scheduled reports, inform them to reschedule all reports if their passwords have been changed.

Determining whether to import LDAP users into MicroStrategy

To connect your LDAP users and groups to users and groups in MicroStrategy, you can either import the LDAP users and groups into the MicroStrategy metadata or you can simply create a link between users and groups in the LDAP directory and in MicroStrategy. Importing a user creates a new user in MicroStrategy based on an existing user in the LDAP directory. Linking a user connects an LDAP user's information to an existing user in MicroStrategy. You can also allow LDAP users to log in to the MicroStrategy system anonymously, without an associated MicroStrategy user. The benefits and considerations of each method are described in the table below.

Connection Type	Benefits	Considerations
Import LDAP users and groups	• Users and groups are created in the metadata. • Users and groups can be assigned additional privileges and permissions in MicroStrategy. • Users have their own inboxes and personal folders in MicroStrategy.	• In environments that have many LDAP users, importing can quickly fill the metadata with these users and their related information. • Users and groups may not have the correct permissions and privileges when they are initially imported into MicroStrategy.

© 2016, MicroStrategy Inc.

Connection Type	Benefits	Considerations
Link users and groups without importing	• For environments that have many LDAP users, linking avoids filling the metadata with users and their related information. • You can use Command Manager to automate the linking process using scripts. See the Command Manager Help for details.	• Users to be linked to must already exist in the MicroStrategy metadata.
Allow anonymous or guest users	• Users can log in immediately without having to create a new MicroStrategy user.	• Privileges are limited to those for the Public/Guest group. • Users' personal folders and Inboxes are deleted from the system after they log out.

The options for importing users into MicroStrategy are described in detail in the following sections:

- *Importing LDAP users and groups into MicroStrategy, page 119*
- *Linking users and groups without importing, page 120*
- *Allowing anonymous/guest users with LDAP authentication, page 121*

You can modify your import settings at any time, for example, if you choose not to import users initially, but want to import them at some point in the future. The steps to modify your LDAP settings are described in *Managing LDAP authentication, page 125*.

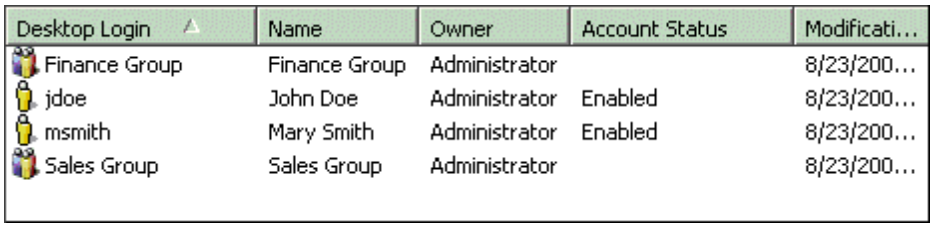
Importing LDAP users and groups into MicroStrategy

You can choose to import LDAP users and groups at login, in a batch process, or a combination of the two. Imported users are automatically members of MicroStrategy's LDAP Users group, and are assigned the access control list (ACL) and privileges of that group. To assign different ACLs or privileges to a user, you can move the user to another MicroStrategy user group.

When an LDAP user is imported into MicroStrategy, you can also choose to import that user's LDAP groups. If a user belongs to more than one group, all the user's groups are imported and created in the metadata. Imported LDAP groups are created within MicroStrategy's LDAP Users folder and in MicroStrategy's User Manager.

LDAP users and LDAP groups are all created within the MicroStrategy LDAP Users group at the same level. While the LDAP relationship between a user and any associated groups exists in the MicroStrategy metadata, the relationship is not visually represented in Developer. For example, looking in the LDAP Users folder in MicroStrategy immediately after an import or synchronization, you might see the following list of imported LDAP users and groups:

© 2016, MicroStrategy Inc.



Desktop Login	Name	Owner	Account Status	Modificati...
Finance Group	Finance Group	Administrator		8/23/200...
jdoe	John Doe	Administrator	Enabled	8/23/200...
msmith	Mary Smith	Administrator	Enabled	8/23/200...
Sales Group	Sales Group	Administrator		8/23/200...

If you want a users' groups to be shown in MicroStrategy, you must manually move them into the appropriate groups.

The relationship between an imported LDAP user or group and the MicroStrategy user or group is maintained by a link in the MicroStrategy metadata, which is in the form of a Distinguished Name. A *Distinguished Name (DN)* is the unique identifier of an entry (in this case a user or group) in the LDAP directory.

The MicroStrategy user's Distinguished Name is different from the DN assigned for the authentication user. The authentication user's DN is the DN of the MicroStrategy account that is used to connect to the LDAP server and search the LDAP directory. The authentication user can be anyone who has search privileges in the LDAP server, and is generally the LDAP administrator.

Removing a user from the LDAP directory does not effect the user's presence in the MicroStrategy metadata. Deleted LDAP users are not automatically deleted from the MicroStrategy metadata during synchronization. You can revoke a user's privileges in MicroStrategy, or remove the user manually.

You cannot export users or groups from MicroStrategy to an LDAP directory.

Linking users and groups without importing

A link is a connection between an LDAP user or group and a MicroStrategy user or group which allows an LDAP user to log in to MicroStrategy. Unlike an imported LDAP user, a linked LDAP user is not created in the MicroStrategy metadata.

An LDAP group can only be linked to a MicroStrategy group, and an LDAP user can only be linked to a MicroStrategy user. It is not possible to link a group to a user without giving the user membership in the group.

When an LDAP user or group is linked to an existing MicroStrategy user or group, no new user or group is created within the MicroStrategy metadata as with importing. Instead, a link is established between an existing MicroStrategy user or group and an LDAP user or group, which allows the LDAP user to log in to MicroStrategy.

The link between an LDAP user or group and the MicroStrategy user or group is maintained in the MicroStrategy metadata in the form of a shared Distinguished Name.

The user's or group's LDAP privileges are not linked with the MicroStrategy user. In MicroStrategy, a linked LDAP user or group receives the privileges of the MicroStrategy user or group to which it is linked.

LDAP groups cannot be linked to MicroStrategy user groups. For example, you cannot link an LDAP group to MicroStrategy's Everyone group. However, it is possible to link an LDAP user to a MicroStrategy user that has membership in a MicroStrategy group.

© 2016, MicroStrategy Inc.

Allowing anonymous/guest users with LDAP authentication

An anonymous or guest LDAP user in MicroStrategy is an existing LDAP user who logs in to MicroStrategy using any login and password that is not imported into MicroStrategy or linked to a MicroStrategy user.

LDAP guest users can use MicroStrategy's capabilities and be authenticated as LDAP guests. All LDAP guest users receive the privileges of the LDAP Public/Guest and the Public/Guest groups in MicroStrategy.

The LDAP server must be configured to allow anonymous or guest authentication requests from MicroStrategy.

Because guest users are not present in the metadata, there are certain actions these users cannot perform in MicroStrategy, even if the associated privileges and permissions are explicitly assigned. Examples include most administrative actions.

When the user is logged in as an anonymous/guest user:

- The user does not have a History List, because the user is not physically present in the metadata.
- The user cannot create objects and cannot schedule reports.
- The User Connection monitor records the LDAP user's user name.
- Intelligence Server statistics record the session information under the user name LDAP USER.

Determining whether to automatically synchronize LDAP user and group information

In any company's security model, steps must be taken to account for a changing group of employees. Adding new users and removing ones that are no longer with the company is straightforward. Accounting for changes in a user's name or group membership can prove more complicated. To ease this process, MicroStrategy supports user name/login and group synchronization with the information contained within an LDAP directory.

If you choose to have MicroStrategy automatically synchronize LDAP users and groups, any LDAP group changes that have occurred within the LDAP server will be applied within MicroStrategy the next time an LDAP user logs in to MicroStrategy. This keeps the LDAP directory and the MicroStrategy metadata in synchronization.

By synchronizing users and groups between your LDAP server and MicroStrategy, you can update the imported LDAP users and groups in the MicroStrategy metadata with the following modifications:

- **User synchronization**:
  - User details such as user name in MicroStrategy are updated with the latest definitions in the LDAP directory.
- **Group synchronization**:

© 2016, MicroStrategy Inc.

    - Group details such as group name in MicroStrategy are updated with the latest definitions in the LDAP directory.

When synchronizing LDAP users and groups in MicroStrategy, you should be aware of the following circumstances:

- If an LDAP user or group has been given new membership to a group that has not been imported or linked to a group in MicroStrategy and import options are turned off, the group cannot be imported into MicroStrategy and thus cannot apply its permissions in MicroStrategy.

  For example, User1 is a member of Group1 in the LDAP directory, and both have been imported into MicroStrategy. Then, in the LDAP directory, User1 is removed from Group1 and given membership to Group2. However, Group2 is not imported or linked to a MicroStrategy group. Upon synchronization, in MicroStrategy, User1 is removed from Group1, and is recognized as a member of Group2. However, any permissions for Group2 are not applied for the user until Group2 is imported or linked to a MicroStrategy group. In the interim, User1 is given the privileges and permissions of the LDAP Users group.

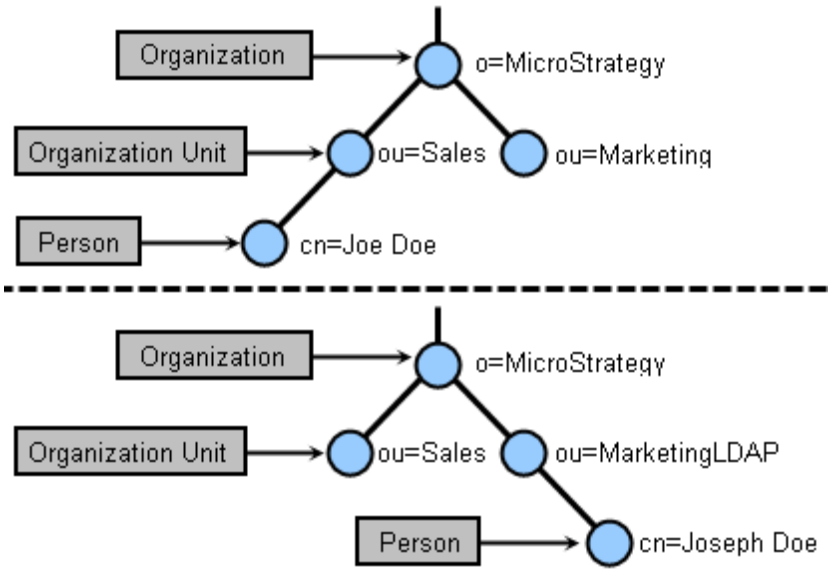
- When users and groups are deleted from the LDAP directory, the corresponding MicroStrategy users and groups that have been imported from the LDAP directory remain in the MicroStrategy metadata. You can revoke users' and groups' privileges in MicroStrategy and remove the users and groups manually.

- Regardless of your synchronization settings, if a user's password is modified in the LDAP directory, a user must log in to MicroStrategy with the new password. LDAP passwords are not stored in the MicroStrategy metadata. MicroStrategy uses the credentials provided by the user to search for and validate the user in the LDAP directory.

Consider a user named Joe Doe who belongs to a particular group, Sales, when he is imported into MicroStrategy. Later, he is moved to a different group, Marketing, in the LDAP directory. The LDAP user Joe Doe and LDAP groups Sales and Marketing have been imported into MicroStrategy. Finally, the user name for Joe Doe is changed to Joseph Doe, and the group name for Marketing is changed to MarketingLDAP.

The images below show a sample LDAP directory with user Joe Doe being moved within the LDAP directory from Sales to Marketing.

© 2016, MicroStrategy Inc.



The following table describes what happens with users and groups in MicroStrategy if users, groups, or both users and groups are synchronized.

Sync Users?	Sync Groups?	User Name After Synchronization	Group Name After Synchronization	User Membership After Synchronization
No	No	Joe Doe	Marketing	Sales
No	Yes	Joe Doe	MarketingLDAP	Sales
Yes	No	Joseph Doe	Marketing	Marketing
Yes	Yes	Joseph Doe	MarketingLDAP	MarketingLDAP

Setting up LDAP authentication in MicroStrategy

When you have collected the connection information for your LDAP server and your LDAP SDK, you can use the LDAP Connectivity Wizard to set up your LDAP connection. The LDAP Connectivity Wizard helps step you through the initial setup of using your LDAP server to authenticate users and groups in MicroStrategy. The steps to set up your LDAP connection are described below.

Prerequisites

- You have collected the information for your LDAP server, and made decisions regarding the LDAP authentication methods you want to use, as described in *Checklist: Information required for connecting your LDAP server to MicroStrategy, page 107*.
- If you want Intelligence Server to access your LDAP server over a secure SSL connection, you must do the following:

a Obtain a valid certificate from your LDAP server and save it on the machine where Intelligence Server is installed. The steps to obtain the certificate depend on your LDAP vendor, and the operating system that your LDAP server runs on. For specific steps, refer to the documentation for your LDAP vendor.

b Follow the procedure recommended by your operating system to install the certificate.

To set up LDAP authentication in MicroStrategy

Connecting your LDAP server using the LDAP Connectivity Wizard

1. In Developer, log in to a project source, as a user with administrative privileges.
2. From the **Administration** menu, select **Server**, and click **LDAP Connectivity Wizard**. The LDAP Connectivity Wizard opens.
3. On the Welcome page, click **Next**. The LDAP Server Information page opens.
4. Type the following information:
   - **Host**: The machine name or IP address of the LDAP server.
   - **Port**: The network port that the LDAP server uses. For clear text connections, the default value is `389`. If you want Intelligence Server to access your LDAP over an encrypted SSL connection, the default value is `636`.
5. If you want Intelligence Server to access your LDAP over an encrypted SSL connection, select **SSL (encrypted)**. The Server Certificate file field is enabled.
6. In the **Server Certificate file** field, depending on your LDAP server vendor, point to the SSL certificate in the following ways:
   - **Microsoft Active Directory**: No information is required.
   - **Sun ONE/iPlanet**: Provide the path to the certificate. Do not include the file name.
   - **Novell**: Provide the path to the certificate, including the file name.
   - **IBM**: Use Java GSKit 7 to import the certificate, and provide the key database name with full path, starting with the home directory.
   - **Open LDAP**: Provide the path to the directory that contains the CA certificate file `cacert.pem`, the server certificate file `servercrt.pem`, and the server certificate key file `serverkey.pem`.
   - **HP-UX**: Provide the path to the certificate. Do not include the file name.
7. Click **Next**. The Platform Connectivity page opens.
8. Enter the details of your LDAP SDK, and click **Next**.

 © 2016, MicroStrategy Inc.

9 Step through the LDAP Connectivity Wizard to enter the remaining information, such as the LDAP search filters to use to find users, whether to import users into MicroStrategy, and so on. For help on specific options in the wizard, click **Help** to open the *System Administrator Help*.

10 When you have entered all the information, click **Finish** to exit the LDAP Connectivity Wizard. You are prompted to test the LDAP connection. It is recommended that you test the connection to catch any errors with the connection parameters you have provided.

Enabling LDAP authentication for your project source

11 In the Folder List, right-click the project source, and select **Modify Project Source**. The Project Source Manager opens.

12 On the **Advanced** tab, select **Use LDAP Authentication**.

13 Click **OK** to accept your changes and close the Project Source Manager.

Enabling LDAP authentication for MicroStrategy Web

14 From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Web Administrator**. The Web Administrator Page opens in a web browser.

15 On the left, under **Intelligence Server**, select **Default Properties**.

16 In the **Login** area, for **LDAP Authentication**, select the **Enabled** check box.

17 If you want LDAP authentication to be the default authentication mode for MicroStrategy Web, for LDAP Authentication, select the **Default** option.

If your environment includes multiple Intelligence Servers connected to one MicroStrategy Web server, users are authenticated to all the Intelligence Servers using their LDAP credentials, and then shown a list of projects they can access. However, if one or more of the Intelligence Servers does not use LDAP authentication, the projects for those servers may not be displayed. To avoid this scenario, in the **Project list** drop-down menu, ensure that **Show all the projects connected to the Web Server before the user logs in** is selected.

18 Click **Save**.

Managing LDAP authentication

While working with MicroStrategy and implementing LDAP authentication, you may want to improve performance or troubleshoot your LDAP implementation. The sections below cover steps that can help your LDAP authentication and MicroStrategy systems work as a cohesive unit.

- If your LDAP server information changes, or to edit your LDAP authentication settings in general, see *Modifying your LDAP authentication settings, page 126*.

- If you want to modify the settings for importing users into MicroStrategy, for example, if you initially chose not to import users, and now want to import users and groups, see *Importing LDAP users and groups into MicroStrategy, page 127*.
- If you choose to synchronize users and groups in batches, and want to select a synchronization schedule, see *Selecting schedules for importing and synchronizing users, page 130*.
- If you are using single sign-on (SSO) authentication systems, such as Windows NT authentication or trusted authentication, you can link users' SSO credentials to their LDAP user names, as described in *Using LDAP with single sign-on authentication systems, page 130*.
- Depending on the way your LDAP directory is configured, You can import additional LDAP attributes for users, for example, a `countryCode` attribute, indicating the user's location. These additional LDAP attributes can be used to create security filters for users, such as displaying data that is relevant to the user's country. For information on creating these security filters, see *Using LDAP attributes in security filters, page 131*.

Modifying your LDAP authentication settings

Depending on changes in your organization's policies, you may need to modify the LDAP authentication settings in MicroStrategy. To modify your LDAP authentication settings, you can use the Intelligence Server Configuration Editor. The steps to access the LDAP settings in the Intelligence Server Configuration Editor are described below.

To access LDAP authentication settings in the Intelligence Server Configuration Editor

1. In Developer, log in to a project source as a user with administrative privileges.
2. From the **Administration** menu, select **Server**, and click **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **LDAP** category. The LDAP settings are displayed. You can modify the following:
   - Your LDAP server settings, such as the machine name, port, and so on.
   - Your LDAP SDK information, such as the location of the LDAP SDK DLL files.
   - The LDAP search filters that Intelligence Server uses to find and authenticate users.
   - If you are importing and synchronizing users or groups in batches, the synchronization schedules.
   - If you are importing users and groups, the import settings.
4. For help on specific options in the LDAP category, click **Help** to open the *System Administrator Help*.

 © 2016, MicroStrategy Inc.

Importing LDAP users and groups into MicroStrategy

You can choose to import LDAP users and groups at login, in a batch process, or a combination of the two, described as follows:

- Importing users and groups at login: When an LDAP user logs in to MicroStrategy for the first time, that user is imported into MicroStrategy and a physical MicroStrategy user is created in the MicroStrategy metadata. Any groups associated with that user that are not already in MicroStrategy are also imported and created in the metadata.
- Importing users and groups in batches: The list of users and groups are returned from user and group searches on your LDAP directory. MicroStrategy users and groups are created in the MicroStrategy metadata for all imported LDAP users and groups.

This section covers the following:

- For information on setting up user and group import options, see *Importing users and groups into MicroStrategy, page 127*.
- Once you have set up user and group import options, you can import additional LDAP information, such as users' email addresses, or specific LDAP attributes. For steps, see *Importing users' email addresses, page 129*.
- For information on assigning security settings after users are imported, see *User privileges and security settings after import, page 129*.

Importing users and groups into MicroStrategy

You can choose to import users and their associated groups when a user logs in to MicroStrategy for the first time.

Prerequisites

- Ensure that you have reviewed the information and made decisions regarding your organization's policy on importing and synchronizing user information, described in the following sections:
  - *Determining whether to import LDAP users into MicroStrategy, page 118*
  - *Determining whether to automatically synchronize LDAP user and group information, page 121*
- If you want to import users and groups in batches, you must define the LDAP search filters to return lists of users and groups to import into MicroStrategy. For information on defining search filters, see *Defining LDAP search filters to verify and import users and groups at login, page 112*.

To import users and/or groups into MicroStrategy

1. In Developer, log in to a project source as a user with administrative privileges.

2. From the **Administration** menu, select **Server**, and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **LDAP** category, then expand **Import**, and then select **Import/Synchronize**.
4. If you want to import user and group information when users log in, in the Import/Synchronize at Login area, do the following:
   - To import users at login, select **Import Users**.
   - To allow MicroStrategy's user information to automatically synchronize with the LDAP user information, select **Synchronize MicroStrategy User Login/User Name with LDAP**.
   - To import groups at login, select the **Import Groups**.
   - To allow MicroStrategy's group information to automatically synchronize with the LDAP group information, select **Synchronize MicroStrategy Group Name with LDAP**.
5. If you want to import user and group information in batches, in the Import/Synchronize in Batch area, do the following:
   - To import users in batches, select **Import Users**. You must also enter a user search filter in the **Enter search filter for importing list of users** field to return a list of users to import.
   - To synchronize MicroStrategy's user information with the LDAP user information, select **Synchronize MicroStrategy User Login/User Name with LDAP**.
   - To import groups in batches, select **Import Groups**. You must also enter a group search filter in the **Enter search filter for importing list of groups** field to return a list of users to import.
   - To synchronize MicroStrategy's group information with the LDAP group information, select **Synchronize MicroStrategy Group Name with LDAP**.
6. To modify the way that LDAP user and group information is imported, for example, to import group names as the LDAP distinguished name, under the LDAP category, under Import, click **User/Group**. For specific information on the options available, click **Help** to open the *System Administrator Help*.
7. To save your changes and close the Intelligence Server Configuration Editor, click **OK**.

Once a user or group is created in MicroStrategy, the users are given their own inboxes and personal folders. Additionally, you can do the following:

- Import users' email addresses. For steps, see *Importing users' email addresses, page 129*.
- Assign privileges and security settings that control what a user can access in MicroStrategy. For information on assigning security settings after users are imported, see *User privileges and security settings after import, page 129*.

 © 2016, MicroStrategy Inc.

- Import additional LDAP attributes, which can then be used in security filters for users. For steps, see *Using LDAP attributes in security filters, page 131*.

Importing users' email addresses

Depending on your requirements, you can import additional information, such as users' email addresses, from your LDAP directory. For example, If you have a license for MicroStrategy Distribution Services, then when you import LDAP users, either in a batch or at login, you can import these email addresses as contacts associated with those users. For information about Distribution Services, see *Overview of Distribution Services, page 501*.

MicroStrategy 9 imports the primary email address for each LDAP user.

To import users' email addresses from LDAP

1. In Developer, log in to a project source as a user with administrative privileges.
2. From the **Administration** menu, select **Server**, and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **LDAP** category, then expand **Import**, and select **Options**.
4. Select **Import Email Address**.
5. Select whether to use the default LDAP email address attribute of `mail`, or to use a different attribute. If you want to use a different attribute, specify it in the text field.
6. From the **Device** drop-down list, select the email device that the email addresses are to be associated with.
7. Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.

User privileges and security settings after import

A user's LDAP privileges and security settings are not imported along with a user. Imported users receive the privileges of the MicroStrategy LDAP Users group. You can add additional privileges to specific users in the LDAP Users group using the standard MicroStrategy process in the User Editor. You can also adjust privileges for the LDAP Users group as a whole.

Similarly, a group's LDAP privileges and security settings are not imported along with the group. Group privileges can be modified using the MicroStrategy Group Editor.

The privileges and security settings assigned to LDAP users imported in MicroStrategy depend on the users' associated MicroStrategy group privileges and security permissions. To see the default privileges assigned to a user or group, in the folder list, expand your project source, expand **Administration**, and then expand **User Manager**. Right-click the group (or select the group and right-click the user) and select **Edit**. The Project Access tab

© 2016, MicroStrategy Inc.

displays all privileges for each project in the project source. For details on editing user and group privileges, see the *MicroStrategy Developer Help*.

The process of synchronizing users and groups can modify which groups a user belongs to, and thus modify the user's privileges and security settings.

Selecting schedules for importing and synchronizing users

If you choose to synchronize users and groups in batches, you can select a schedule that dictates when LDAP users and groups are synchronized in MicroStrategy. For information on creating and using schedules, see *Creating and managing schedules, page 479*. To select a synchronization schedule for LDAP, follow the steps below.

To select a schedule for importing and synchronizing users

1. In Developer, log in to a project source as a user with administrative privileges.
2. From the **Administration** menu, select **Server**, and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **LDAP** category, then click **Schedules**. The available schedules are displayed. By default, all the check boxes for all the schedules are cleared.
4. Select the schedules to use as LDAP user and group synchronization schedules.
5. To synchronize your MicroStrategy users and groups with the latest LDAP users and groups immediately, select **Run schedules on save**.
6. Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.

Using LDAP with single sign-on authentication systems

If you are using single sign-on (SSO) authentication systems, such as Windows NT authentication or trusted authentication, you can link users' SSO credentials to their LDAP user names, and import the LDAP user and group information into MicroStrategy. For information about configuring a single sign-on system, see *Enabling single sign-on authentication, page 132*.

Depending on the SSO authentication system you are using, refer to one of the following sections for steps:

- If you are using Windows NT authentication, see *Linking a Windows login to an LDAP user, page 137*.
- If you are using integrated or trusted authentication, see *Linking integrated authentication users to LDAP users, page 142*.

 © 2016, MicroStrategy Inc.

Using LDAP attributes in security filters

You may want to integrate LDAP attributes into your MicroStrategy security model. For example, you want users to only see sales data about their country. You import the LDAP attribute `countryName`, create a security filter based on that LDAP attribute, and then you assign that security filter to all LDAP users. Now, when a user from Brazil views a report that breaks down sales revenue by country, she only sees the sales data for Brazil.

LDAP attributes are imported into MicroStrategy as system prompts. A system prompt is a special type of prompt that is answered automatically by Intelligence Server. The LDAP attribute system prompts are answered with the related LDAP attribute value for the user who executes the object containing the system prompt. You import LDAP attributes into MicroStrategy from the Intelligence Server Configuration Editor.

Once you have created system prompts based on your LDAP attributes, you can use those system prompts in security filters to restrict the data that your users can see based on their LDAP attributes. For information about using system prompts in security filters, including instructions, see *Using a single security filter for multiple users: System prompts, page 89*. For general information about security filters, see *Restricting access to data: Security filters, page 79*.

To import an LDAP attribute into a project

1. In Developer, log into a project source.
2. From the **Administration** menu, point to **Server** and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **LDAP** category, then expand the **Import** category, and then select **Attributes**.
4. From the **Select LDAP Attributes** drop-down list, select the LDAP attribute to import.
5. From the **Data Type** drop-down list, select the data type of that attribute.
6. Click **Add**. A system prompt is created based on that attribute. You can now use that system prompt in filters and security filters.
7. Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.

Controlling project access with LDAP attributes

By default, an LDAP user can log in to a project source even if the LDAP attributes that are used in system prompts are not defined for that user. To increase the security of the system, you can prevent LDAP users from logging in to a project source if all LDAP attributes that are used in system prompts are not defined for that user.

When you select this option, you prevent all LDAP users from logging in to the project source if they do not have all the required LDAP attributes. This affects all users using

© 2016, MicroStrategy Inc.

LDAP authentication, and also any users using Windows, Trusted, or Integrated authentication if those authentication systems have been configured to use LDAP. For example, if you are using Trusted authentication with a SiteMinder single sign-on system, and SiteMinder is configured to use an LDAP directory, this option prevents SiteMinder users from logging in if they do not have all the required LDAP attributes.

Be aware of the following:

- This setting prevents users from logging in to all projects in a project source.
- If your system uses multiple LDAP servers, make sure that all LDAP attributes used by Intelligence Server are defined on all LDAP servers. If a required LDAP attribute is defined on LDAP server A and not on LDAP server B, and the **User login fails if LDAP attribute value is not read from the LDAP server** check box is selected, users from LDAP server B will not be able to log in to MicroStrategy.

To only allow users with all required LDAP attributes to log in to the system

1. In Developer, log into a project source.
2. From the **Administration** menu, point to **Server** and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **LDAP** category, then expand the **Import** category, and then select **Attributes**.
4. Select the **User logon fails if LDAP attribute value is not read from the LDAP server** check box.
5. Click **OK** to accept your changes and close the Intelligence Server Configuration Editor. You may need to restart Intelligence Server for this change to take effect.

Troubleshooting

There may be situations where you can encounter problems or errors while trying to integrate MicroStrategy with your LDAP directory. For troubleshooting information and procedures, see *Troubleshooting LDAP authentication, page 757*.

Enabling single sign-on authentication

Enabling authentication to several programs using a single login is known as single sign-on authentication. There are three different types of authentication that can be used to

 © 2016, MicroStrategy Inc.

enable single sign-on in MicroStrategy. These types of authentication are:

- Windows — *Implementing Windows NT authentication, page 133*
- Integrated - *Enabling integrated authentication, page 140*
- Third-party authentication - *Enabling Single Sign-on to Web, Mobile, and Office with third-party authentication, page 161*

Implementing Windows NT authentication

If you use Windows 2003 as your network operating system and your users are already defined in a Windows 2003 directory, then you can enable Windows authentication in MicroStrategy to allow users access without having to enter their login information.

The Apple Safari web browser does not support Windows authentication with MicroStrategy Web.

Use the procedures in the rest of this section to enable single sign-on with Windows authentication in MicroStrategy Web. For high-level steps to configure these settings, see *Steps to enable single sign-on to MicroStrategy Web using Windows authentication, page 134*.

To use Windows authentication you must create users in the MicroStrategy environment and then link them to Windows users. Linking enables Intelligence Server to map a Windows user to a MicroStrategy user. See *Linking a Windows domain user to a MicroStrategy user, page 137*.

You can also create MicroStrategy users from existing Windows by importing either user definitions or group definitions. For more information on importing users or groups, see the *MicroStrategy Developer Help*.

To use Windows authentication with MicroStrategy Web, you must be running MicroStrategy Web or Web Universal under Microsoft IIS. Non-IIS web servers do not support Windows authentication. See *Enabling integrated authentication for IIS, page 147*.

If the Windows domain account information is linked to a MicroStrategy user definition, a MicroStrategy Web user can be logged in automatically through MicroStrategy Web. When a user accesses MicroStrategy Web, IIS detects the Windows user and sends the login information to Intelligence Server. If the Windows user is linked to a MicroStrategy user, Intelligence Server starts a session for that user. For information on setting up MicroStrategy Web to allow single sign-on using Windows authentication, see *Enabling Windows authentication login for MicroStrategy Web, page 139*.

Enabling Windows authentication in MicroStrategy Web to allow single sign-on

Single sign-on authentication allows users to type their login credentials once, and have access to multiple software applications securely, because the system can apply that single authentication request to all the applications that the user need access to. It is possible to use Windows authentication to enable single sign-on for MicroStrategy Web.

© 2016, MicroStrategy Inc.

There are several configurations that you must make to enable Windows authentication in MicroStrategy Web. To properly configure MicroStrategy Web, Microsoft Internet Information Services (IIS), and the link between Microsoft and MicroStrategy users, follow the procedure *Steps to enable single sign-on to MicroStrategy Web using Windows authentication, page 134*.

Steps to use Windows authentication with Microsoft Sharepoint and MicroStrategy Web are in the MicroStrategy Developer Library (MSDL). The MicroStrategy SDK and MSDL contain information on customizing MicroStrategy Web.

Prerequisites

Before continuing with the procedures described in the rest of this section, you must first set up a Windows domain that contains a domain name for each user that you want to allow single sign-on access to MicroStrategy Web with Windows authentication.

In addition, you must be connected to the MicroStrategy Web machine without a proxy. Windows authentication does not work over a proxy connection. For more information, including some possible work-arounds, see Microsoft's IIS documentation.

Steps to enable single sign-on to MicroStrategy Web using Windows authentication

1. Enable integrated Windows authentication for Microsoft IIS. See *Enabling Windows authentication for Microsoft IIS, page 134*.
2. If you are using MicroStrategy Web Universal on a J2EE-based application server such as Apache Tomcat, enable the MicroStrategy ISAPI filter in IIS, to support Windows authentication. For steps, see *Enabling Windows authentication for J2EE-based application servers, page 135*.
3. Create a link between a Windows domain user and a MicroStrategy Web user for each person that will be accessing MicroStrategy Web with Windows authentication. See *Linking a Windows domain user to a MicroStrategy user, page 137*.
4. Define a project source to use Windows authentication. See *Defining a project source to use Windows authentication, page 138*.
5. Enable Windows authentication in MicroStrategy Web. See *Enabling Windows authentication login for MicroStrategy Web, page 139*.
6. Configure each MicroStrategy Web user's browser for single sign-on. See *Configuring a browser for single sign-on to MicroStrategy Web, page 140*.

Enabling Windows authentication for Microsoft IIS

Microsoft Internet Information Services is an Internet server that is integral to Windows authentication. You must configure IIS to enable Windows authentication in the MicroStrategy virtual directory to support integrated authentication to MicroStrategy Web.

 © 2016, MicroStrategy Inc.

The steps to perform this configuration are provided in the procedure below, which may vary depending on your version of IIS. The following links can help you find information on how to enable integrated authentication for your version of IIS:

- **IIS 7**: See *http://technet.microsoft.com/en-us/library/cc754628(WS.10).aspx* for information on enabling Windows authentication for IIS 7.

  If you are using IIS 7 on Windows Server 2008, ensure the following:

  - The MicroStrategyWebPool application pool is started, and the Managed Pipeline is set to **Integrated**.
  - ASP.NET Impersonation is enabled. For information on enabling ASP.NET Impersonation in IIS 7, see *http://technet.microsoft.com/en-us/library/cc730708(WS.10).aspx*.

- **IIS 6**: See *http://technet.microsoft.com/en-us/library/cc780160(WS.10).aspx* for information on enabling Windows authentication for IIS 6.
- **IIS 5**: See *http://support.microsoft.com/kb/215383* for information on enabling Windows authentication for IIS 5.

The third-party products discussed below are manufactured by vendors independent of MicroStrategy, and the information provided is subject to change. Refer to the appropriate third-party vendor documentation for updated IIS support information.

To enable Windows authentication in Microsoft IIS

1. On the MicroStrategy Web server machine, access the **IIS Internet Service Manager**.
2. Navigate to and right-click the MicroStrategy virtual folder, and select **Properties**.
3. Select the **Directory Security** tab, and then under **Anonymous access and authentication control**, click **Edit**. The **Authentication Methods** dialog box opens.
4. Clear the **Anonymous access** check box.
5. Select the **Integrated Windows authentication** check box.
6. Click **OK** to save your changes and close the Authentication Methods dialog box.
7. Click **OK** again to save your changes to the MicroStrategy virtual folder.
8. Restart IIS for the changes to take effect.

Enabling Windows authentication for J2EE-based application servers

If you use a J2EE-compliant application server other than IIS to deploy MicroStrategy Web Universal, you must configure IIS to share users' Windows authentication

credentials with your application server. To allow IIS to share users' Windows authentication credentials with your applications server, you must add the MicroStrategy ISAPI filter to IIS, as described in the steps below.

The third-party products discussed below are manufactured by vendors independent of MicroStrategy, and the information provided is subject to change. Refer to the appropriate third-party vendor documentation for the latest information.

Prerequisites

- In your MicroStrategy installation folder, locate the `MBWBAUTH.dll` file. By default, the file is located in `C:\Program Files (x86)\Common Files\MicroStrategy`.

Depending on the version of IIS you are using, refer to one of the following procedures to enable the MicroStrategy ISAPI filter:

- *To enable the MicroStrategy ISAPI filter in IIS 6, page 136*
- *To enable the MicroStrategy ISAPI filter in IIS 7, page 136*

To enable the MicroStrategy ISAPI filter in IIS 6

1. In IIS, right-click the default web site, and select **Properties**. The Default Web Site Properties dialog box opens.
2. Click the **ISAPI Filters** tab. A list of ISAPI filters for your IIS installation is shown.
3. Click **Add**. An Open File dialog box is displayed.
4. Browse to the location of the `MBWBAUTH.dll` file. By default, the file is located in `C:\Program Files (x86)\Common Files\MicroStrategy`.
5. Select `MBWBAUTH.dll` and click **OK**. The `MBWBAUTH` ISAPI filter is added to the list of ISAPI filters.
6. Restart your IIS server.

To enable the MicroStrategy ISAPI filter in IIS 7

1. In IIS, select the default web site. The Default Web Site Home page is shown.
2. In the Default Web Site Home page, double-click **ISAPI Filters**. A list of ISAPI filters for your IIS installation is shown.
3. In the Actions pane, click **Add**. The Add ISAPI Filter dialog box opens.
4. In the **Filter name** field, type a name for the filter. For example, `MicroStrategy ISAPI Filter`.
5. Next to the Executable field, click Browse (**...**). An Open File dialog box opens.

 © 2016, MicroStrategy Inc.

6. Browse to the location of the `MBWBAUTH.dll` file. By default, the file is located in `C:\Program Files (x86)\Common Files\MicroStrategy`.
7. Select `MBWBAUTH.dll` and click **OK**. The path of the file is shown in the Executable field.
8. Click **OK**. The ISAPI filter is added to IIS.
9. Restart your IIS server.

Linking a Windows domain user to a MicroStrategy user

Once IIS has been configured to allow integrated Windows authentication, a link must be created between a user's MicroStrategy user name and the user's Windows domain user name. The required steps are detailed below.

To link a Windows domain user to a MicroStrategy user

1. In Developer, log in to a project source using an account with administrative privileges.
2. From the Folder List, expand a project source, then expand **Administration**, and then expand **User Manager**.
3. Navigate to the MicroStrategy user you want to link a Windows user to. Right-click the MicroStrategy user and select **Edit**. The User Editor opens.
4. Expand **Authentication**, then select **Metadata**.
5. Under **Windows Authentication**, in the **Link Windows user** area, provide the Windows user name for the user you want to link the MicroStrategy user to. There are two ways to do this:
   - Click **Browse** to select the user from the list of Windows users displayed.
   - Click **Search** to search for a specific Windows user by providing the Windows login to search for and, optionally, the Windows domain to search. Then click **OK** to run the search.
6. Click **OK** to save your changes and close the User Editor.

Linking a Windows login to an LDAP user

When using LDAP with MicroStrategy, you can reduce the number of times a user needs to enter the same login and password by linking their Windows system login with their LDAP login used in MicroStrategy.

By creating a link between a Windows system login, an LDAP user, and a MicroStrategy user, a single login into the machine authenticates the user for the machine as well as in MicroStrategy.

© 2016, MicroStrategy Inc.

For example, a user logs in to their Windows machine with a linked LDAP login and password and is authenticated. The user then opens Developer and connects to a project source using Windows authentication. Rather than having to enter their login and password to log in to MicroStrategy, the user's login and password authenticated when logging in to their machine is used to authenticate the user. During this process, the user account and any relevant user groups are imported and synchronized for the user.

Prerequisites

- The LDAP Server is configured as the Microsoft Active Directory Server domain controller, which stores the Windows system login information.

To link a Windows login with LDAP and MicroStrategy

1. In Developer, log in to a project source. You must log in as a user with administrative privileges.
2. From the **Administration** menu, select **Server**, and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **LDAP** category, then expand **Import**, and then select **Options**. The Import Options are displayed.
4. Select the **Synchronize user/group information with LDAP during Windows authentication and import Windows link during Batch Import** check box.
5. Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.

Defining a project source to use Windows authentication

For MicroStrategy Web users to gain access to a project in a specific project source using Windows authentication, the project source must first be configured have Windows authentication enabled. The steps for enabling this configuration are detailed below.

To define a project source to use Windows authentication

1. In Developer, log in to a project source using an account with administrative privileges.
2. Right-click the project source and select **Modify Project Source**. The Project Source Manager opens.
3. On the **Advanced** tab, select the **Use network login id (Windows authentication)** option.
4. Click **OK**. The Project Source Manager closes.

 © 2016, MicroStrategy Inc.

Enabling Windows authentication login for MicroStrategy Web

There are two ways to enable access to MicroStrategy Web using Windows authentication. Access can be enabled for the MicroStrategy Web application as a whole, or it can be enabled for individual projects at the project level.

For steps to enable Windows authentication for all of MicroStrategy Web, see *To enable Windows authentication login for MicroStrategy Web, page 139*.

For steps to enable Windows authentication for a project, see *To enable Windows authentication login for a project, page 139*.

To enable Windows authentication login for MicroStrategy Web

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Web Administrator**. The Web Administrator Page opens in a web browser.
2. On the left, under **Intelligence Server**, select **Default Properties**.
3. In the **Login** area, for **Windows Authentication**, select the **Enabled** check box.

   If you want Windows authentication to be the default login mode for MicroStrategy Web, for Windows Authentication, select the Default option.
4. Click **Save**.

To enable Windows authentication login for a project

1. Log into a MicroStrategy Web project as a user with administrative privileges.
2. At the upper left of the page, click the **MicroStrategy** icon, and select **Preferences**. The Preferences page opens.
3. On the left, select **Project Defaults,** then **Security.**
4. In the **Login modes** area, for **Windows Authentication**, select the **Enabled** check box.

   If you want Windows authentication to be the default login mode for this project in MicroStrategy Web, also select the **Default** option.
5. Next to **Apply**, choose whether to apply these settings to all projects, or just to the one you are currently logged into.
6. Click **Apply.**

© 2016, MicroStrategy Inc.

Configuring a browser for single sign-on to MicroStrategy Web

If a MicroStrategy Web user plans to use single sign-on to log in to MicroStrategy Web, each user's browser must be configured to enable integrated authentication. The process to enable integrated authentication is different depending on the browser they use:

- For Internet Explorer, you must enable integrated authentication for the browser, as well as add the MicroStrategy Web server URL as a trusted site. Depending on your security policy, integrated authentication may be enabled by default for Internet Explorer.
- For Firefox, you must add the MicroStrategy Web server URL as a trusted site. The URL must be listed in the about:config page, in the settings network.negotiate-auth.trusted-uris and network.negotiate- auth.delegation-uris.

Enabling integrated authentication

Integrated authentication enables a Windows user to log in once to their Windows machine. The user does not need to log in again separately to Developer or MicroStrategy Web. This type of authentication uses Kerberos to validate a user's credentials.

In addition to authenticating users to Developer and MicroStrategy Web, integrated authentication also passes user credentials down to the database server. This allows each user's credentials to be used to return data from the database.

For single sign-on with integrated authentication to work, users must have user names and passwords that are printable, US-ASCII characters. This limitation is expected behavior in Kerberos. This limitation is important to keep in mind when creating a multilingual environment in MicroStrategy.

Required machine configurations for integrated authentication

To support this type of authentication, you must properly configure MicroStrategy, as well as some third-party tools and options. The table below lists the configurations required, and on which machine the configurations must be performed.

The third-party products discussed in the table and sections below are manufactured by vendors independent of MicroStrategy, and the information provided is subject to change. Refer to the appropriate third-party vendor documentation for details on supporting integrated authentication.

Machine	Required Configurations
Machine hosting the domain controller	Configure a Windows domain controller with Microsoft Active Directory: • To allow users created in a domain to use integrated authentication in MicroStrategy, you must clear the **Account is sensitive and cannot be delegated** authentication option for each user. For information on this configuration, see *Configuring a domain controller and users, page 143*. • If Intelligence Server is run as an application with a particular user account, you

 © 2016, MicroStrategy Inc.

Machine	Required Configurations
	must create a user in the domain with the **Account is trusted for delegation** authentication option selected. This user account can then be used to run Intelligence Server as an application. For information on this configuration, see *Trusting Intelligence Server for delegation, page 143*. • If Intelligence Server is run as a service, define the Intelligence Server machine to be trusted for delegation. You can do this by selecting the **Trust computer for delegation** authentication option for the host machine. For information on this configuration, see *Trusting Intelligence Server for delegation, page 143*. • Define the web server for MicroStrategy Web host machine to be trusted for delegation. To do this, select the **Trust computer for delegation** authentication option for the host machine. For information on this configuration, see *Trusting the MicroStrategy Web and Mobile server host for delegation, page 144*.
UNIX/Linux machine hosting Intelligence Server Universal	If you use Intelligence Server Universal hosted on a UNIX/Linux machine, you must install and configure Kerberos 5 on your UNIX/Linux machine. For information on this configuration, see *Configuring Intelligence Server Universal on UNIX/Linux for integrated authentication, page 144*.
Machine hosting Internet Information Services (IIS) or other MicroStrategy Web application server	Enable integrated authentication for IIS, as described in *Enabling integrated authentication for IIS, page 147*.
	To enable single sign-on authentication to MicroStrategy Web, MicroStrategy Mobile, or MicroStrategy Web Services from a Microsoft Windows machine, you must modify a Windows registry setting (`allowtgtsessionkey`). For information on this configuration, see *Enabling session keys for Kerberos security, page 149*.
	If you use Intelligence Server or MicroStrategy Web Services hosted on a Windows machine, you must configure the krb5.ini file. For information on this configuration, see *Configuring the krb5.ini file, page 149*
Machine hosting a J2EE-compliant application server	If you use a J2EE-compliant application server to deploy MicroStrategy Web Universal, MicroStrategy Mobile Server, or MicroStrategy Web Services, you must perform various configurations to enable integrated authentication, as described in *Enabling integrated authentication for J2EE-compliant application servers, page 150*. **Note**: If you are using integrated authentication for MicroStrategy Mobile, your J2EE-compliant application server must use JDK 1.8 or higher.
MicroStrategy Web user's machine	If a MicroStrategy Web user plans to use single sign-on to log in to MicroStrategy Web, the user must configure their browser to enable integrated authentication. For instructions, see *Configuring a browser for single sign-on to MicroStrategy Web, page 156*.

Machine	Required Configurations
Any machine with the required software for the task	In Developer, link a MicroStrategy user to the domain user. For information on this configuration, see *Linking a domain user to a MicroStrategy user, page 156*.
	In Developer, configure a project source to use integrated authentication. For information on this configuration, see *Using integrated authentication for a project source, page 157*.
	In MicroStrategy Web Administrator, configure MicroStrategy Web to include integrated authentication as an authentication option. For information on this configuration, see *Enabling integrated authentication login mode for MicroStrategy Web, page 158*.
	In MicroStrategy Mobile Administrator, create a Mobile configuration to allow your iOS and Android users to log into MicroStrategy Mobile using integrated authentication. For steps to create a Mobile configuration, see the *Mobile Server Help*.
	In addition to authenticating users to Developer and MicroStrategy Web, integrated authentication can also be extended to pass user credentials down to the database server. To support this optional configuration, see *Enabling integrated authentication to data sources, page 159*.

Linking integrated authentication users to LDAP users

When users log in to MicroStrategy using their integrated authentication credentials, their LDAP group memberships can be imported and synchronized.

By default, users' integrated authentication information is stored in the `userPrincipalName` LDAP attribute. If your system stores integrated authentication information in a different LDAP attribute, you can specify the attribute when you configure the import.

Prerequisites

- The LDAP server has been configured, as described in *Setting up LDAP authentication in MicroStrategy, page 123*.
- You have configured the settings for importing users from your LDAP directory, as described in *Importing LDAP users and groups into MicroStrategy, page 127*.

To import LDAP user and group information for integrated authentication users

1. In Developer, log in to a project source. You must log in as a user with administrative privileges.
2. From the **Administration** menu, select **Server**, and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **LDAP** category, then expand **Import**, and then select **Options**. The Import Options are displayed.

 © 2016, MicroStrategy Inc.

4 Select the **Synchronize user/group information with LDAP during Windows authentication and import Windows link during Batch Import** check box.

5 Select the **Batch import Integrated Authentication/Trusted Authentication unique ID** check box. The Use Default LDAP Attribute option is enabled.

6 By default, users' integrated authentication IDs are stored in the `userPrincipalName` LDAP attribute. If your system stores integrated authentication information in a different LDAP attribute, click **Other**, and type the LDAP attribute that contains users' IDs.

7 Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.

Configuring a domain controller and users

To enable users to be authenticated in MicroStrategy using their Windows login credentials, you must configure a Microsoft Active Directory domain controller to apply user authentication and delegation policies. High-level steps to configure Active Directory to work with integrated authentication in MicroStrategy are provided below. Refer to your Microsoft documentation for detailed information on configuring Active Directory.

For users to be authenticated in MicroStrategy using their Windows login, their Windows user accounts must be created in an Active Directory domain and defined to be delegated. This requires that once the account is created, you must clear the **Account is sensitive and cannot be delegated** account option for a user.

Trusting Intelligence Server for delegation

For Intelligence Server to pass login credentials to enable integrated authentication in MicroStrategy, it must be trusted for delegation.

To trust Intelligence Server for delegation, you must perform the following tasks:

- You must create a Service Principal Name (SPN) for Intelligence Server, and map it to the domain user that Intelligence Server runs as. The SPN identifies Intelligence Server as a service that uses Kerberos. For instructions on creating an SPN, refer to the Kerberos documentation.

  If you are running Intelligence Server as a service, the SPN should be in the following format:

  `MSTRSVRSvc/IS_MachineName:ISport`

  If you are running Intelligence Server as an application, the SPN should be in the following format:

  `MSTRSVRSvc/IS_MachineName:ISport@DOMAIN_REALM`

  The formats are explained below:

  - `MSTRSVRSvc`: The Service Class for the Intelligence Server.

    This must be entered exactly as above, with matching case.

- *IS_MachineName*: The fully qualified host name for the machine which is running Intelligence Server.
- *ISPort*: The port where Intelligence Server is hosted.
- *DOMAIN_REALM*: The domain realm of the Intelligence Server, which must be entered in uppercase. It is usually of the form `EXAMPLE.COM`.

The domain realm is required if you are running Intelligence Server as an application. If you are running Intelligence Server as a service, the domain realm is optional.

- In your Active Directory, you must configure the Intelligence Server's domain user to be trusted for delegation, and map the user to this SPN. For example, if the Intelligence Server runs as the user `mstr-iserver`, you must enable the **Account is trusted for delegation** option for the user, and map the user to the SPN.

- Trust Intelligence Server for delegation. For the user account that Intelligence Server runs under, enable the **Account is trusted for delegation** authentication option.

If you are running Intelligence Server as a service, you must also enable the **Trust this computer for delegation to any service (Kerberos only)** option for the Intelligence Server machine.

- Map the Intelligence Server user account to the SPN you created above.

If you are running Intelligence Server Universal on UNIX or Linux, you must perform additional steps on the Intelligence Server machine, as described in *Configuring Intelligence Server Universal on UNIX/Linux for integrated authentication, page 144*.

Trusting the MicroStrategy Web and Mobile server host for delegation

The web server hosts for MicroStrategy Web and MicroStrategy Mobile must be trusted for delegation so that it can pass login credentials to enable integrated authentication in MicroStrategy. You can configure this delegation for the MicroStrategy Web and Mobile server machines in your domain controller. You must select the **Trust this computer for delegation to any service (Kerberos only)** option for the MicroStrategy Web and MicroStrategy Mobile server machines.

Depending on your network, this setting may require a few minutes to take effect.

Configuring Intelligence Server Universal on UNIX/Linux for integrated authentication

If you use Intelligence Server Universal hosted on a UNIX or Linux machine, you must install and configure Kerberos 5 on your UNIX/Linux machine. Configuring Kerberos on your UNIX/Linux machine hosting Intelligence Server Universal enables secure communications to your Windows domain controller.

 © 2016, MicroStrategy Inc.

The configurations listed below are required to configure Intelligence Server Universal with your Windows domain controller and Kerberos security:

- *Install Kerberos 5, page 145*
- *Ensure that the environment variables are set, page 145*
- *Configure the krb5.keytab file for the Intelligence Server, page 146*
- *Configure the krb5.conf file for the Intelligence Server, page 147*

Kerberos only supports US-ASCII characters. Do not use any special characters when installing or configuring Kerberos.

Prerequisites

- Ensure that you have created a Service Principal Name (SPN) for your Intelligence Server, and configured your domain controller to trust Intelligence Server, as described in *Trusting Intelligence Server for delegation, page 143*.
- Ensure that the system clock of the Intelligence Server machine is in sync with the clock on your domain controller.

Install Kerberos 5

You must have Kerberos 5 installed on your UNIX or Linux machine that hosts Intelligence Server Universal. Your UNIX or Linux operating system may come with Kerberos 5 installed. If Kerberos 5 is not installed on your UNIX or Linux machine, refer to the Kerberos documentation for steps to install it.

Ensure that the environment variables are set

Once you have installed Kerberos 5, you must ensure that the following environment variables have been created:

Variable	Description	Default	Required/Optional
${KRB5_HOME}	Location of all Kerberos configuration files	/etc/krb5	Optional
${KRB5_CONFIG}	Location of the default Kerberos configuration file	/etc/krb5/krb5.conf	Required
${KRB5CCNAME}	Location of the Kerberos credential cache	/etc/krb5/krb5_ccache	Optional
${KRB5_KTNAME}	Location of the Kerberos keytab file	/etc/krb5/krb5.keytab	Required

Configure the krb5.keytab file for the Intelligence Server

You must create and configure the `krb5.keytab` file. The steps to configure this file on your UNIX or Linux machine are provided in the procedure below.

Prerequisites

The procedure below requires a few variables to be entered for various commands. This includes information you can gather before you begin the procedure. The required variables in the following procedure are described below:

- *ISMachineName*: The name of the Intelligence Server machine.
- *ISPort*: The port number for Intelligence Server.
- *KeyVersionNumber*: The key version number, retrieved as part of this procedure.
- *EncryptionType*: The encryption type used.

It is recommended that you use `rc4-hmac` as the encryption type. Other encryption types may cause compatibility issues with the Windows Active Directory.

- *DOMAIN_REALM*: The domain realm for your Intelligence Server, which must be entered in uppercase.

To create a krb5.keytab file

1. Log in to your UNIX or Linux machine.
2. Retrieve the key version number for your Intelligence Server service principal name, using the following command:

   ```
   kvno MSTRSVRSvc/ISMachineName:ISPort@DOMAIN_REALM
   ```

   The key version number is displayed on the command line.
3. In the command line, type the following commands:

   ```
   ktutil
   addent -password -p MSTRSVRSvc/
   ISMachineName:ISPort@DOMAIN_REALM -k KeyVersionNumber -
   e EncryptionType
   wkt /etc/krb5/krb5.keytab
   exit
   ```
4. To verify the keytab file, type the following command:

   ```
   kinit -k -t /etc/krb5/krb5.keytab
   MSTRSVRSvc/ISMachineName:ISPort@DOMAIN_REALM
   ```

   The command should run without prompting you for a username and password.

 © 2016, MicroStrategy Inc.

Configure the krb5.conf file for the Intelligence Server

You must create and configure a file named `krb5.conf`. This file is stored in the `/etc/krb5/` directory by default.

If you create a `krb5.conf` file in a directory other than the default, you must update the `KRB5_CONFIG` environment variable with the new location. Refer to your Kerberos documentation for steps to modify the `KRB5_CONFIG` environment variable.

The contents of the `krb5.conf` should be as shown below:

```
[libdefaults]
default_realm = DOMAIN_REALM
default_keytab_name = FILE:/etc/krb5/krb5.keytab
forwardable = true
no_addresses = true

[realms]
DOMAIN_REALM = {
kdc = DC_Address:88
admin_server = DC_Admin_Address:749
}

[domain_realm]
.domain.com = DOMAIN_REALM
domain.com = DOMAIN_REALM
.subdomain.domain.com = DOMAIN_REALM
subdomain.domain.com = DOMAIN_REALM
```

The variables in the syntax above are described below:

- *DOMAIN_REALM*: The domain realm used for authentication purposes. A domain realm is commonly of the form `EXAMPLE.COM`, and must be entered in uppercase.
- *domain.com* and *subdomain.domain.com*: Use this for all domains and subdomains whose users must be authenticated using the default Kerberos realm.
- *DC_Address*: The host name or IP address of the Windows machine that hosts your Active Directory domain controller. This can be the same address as *DC_Admin_Address*.
- *DC_Admin_Address*: The host name or IP address of the Windows machine that hosts your Active Directory domain controller administration server. This can be the same address as *DC_Address*.

Enabling integrated authentication for IIS

Integrated authentication in MicroStrategy requires communication between your Kerberos security system, IIS, and your database.

If you use a J2EE-compliant application server other than IIS to deploy MicroStrategy Web Universal, MicroStrategy Mobile Server, or MicroStrategy Web Services, see *Enabling integrated authentication for J2EE-compliant application servers, page 150*.

You must configure IIS to enable integrated authentication to:

- The MicroStrategy virtual directory to support integrated authentication to MicroStrategy Web, or MicroStrategy Web Services to support MicroStrategy Office. The steps to perform this configuration are provided in the procedure below, which may vary depending on your version of IIS. The following URLs may provide additional information to configure IIS, depending on the version you are using:
  - **IIS 7**: *http://technet.microsoft.com/en-us/library/cc754628(WS.10).aspx*
  - **IIS 6**: *http://technet.microsoft.com/en-us/library/cc780160(WS.10).aspx*
  - **IIS 5**: *http://support.microsoft.com/kb/215383*
- Optionally, the data warehouse. For instructions to enable integrated authentication for the data warehouse, refer to *Enabling integrated authentication to data sources, page 159*.

If you are using Microsoft Analysis Services, to support report subscriptions, you must use connection mapping to pass users' credentials to Analysis Services. For steps to enable connection mapping, see *Connection maps: Standard authentication, connection maps, and partitioned fact tables, page 181*.

To configure IIS to enable integrated authentication to the MicroStrategy virtual directory

1. On the MicroStrategy Web server machine, access the IIS Internet Service Manager.
2. Browse to and right-click the **MicroStrategy** virtual folder and select **Properties**.
3. Select the **Directory Security** tab, and then under **Anonymous access and authentication control**, click **Edit**. The Authentication Methods dialog box opens.
4. Clear the **Enable anonymous access** check box.
5. Select the **Integrated Windows authentication** check box.
6. Click **OK** to save your changes and close the Authentication Methods dialog box.
7. Click **OK** again to save your changes to the MicroStrategy virtual folder.
8. If you want to enable integrated authentication for MicroStrategy Mobile, repeat the above procedure for the **MicroStrategyMobile** virtual folder.
9. If you want to enable integrated authentication for MicroStrategy Web Services, repeat the above procedure for the **MicroStrategyWS** virtual folder.
10. Restart IIS for the changes to take effect.

 © 2016, MicroStrategy Inc.

Creating a Service Principal Name for IIS

It is recommended that you create a Service Principal Name (SPN) for IIS, and map it to the domain user that the application server runs as. The SPN identifies your application server as a service that uses Kerberos. For instructions on creating an SPN, refer to the Kerberos documentation.

The SPN should be in the following format:

`HTTP/ASMachineName`

The format is described below:

- `HTTP`: This is the service class for the application server.
- *`ASMachineName`*: This is the fully qualified host name of the server where the application server is running. It is of the form `machine-name.example.com`.

In your Active Directory, configure the application server's domain user to be trusted for delegation, and map the user to this SPN. For example, if IIS runs as the user `iis`, you must enable the **Account is trusted for delegation** option for the user, and map the user to the SPN. You must also enable the **Trust this computer for delegation to any service (Kerberos only)** option for the machine where IIS is hosted.

Enabling session keys for Kerberos security

To enable single sign-on authentication to MicroStrategy Web from a Microsoft Windows machine, you must modify a Windows registry setting on the machine hosting IIS.

Modification of the `allowtgtsessionkey` registry setting is required by Microsoft to work with Kerberos security. For information on the implications of modifying the registry setting and steps to modify the registry setting, see the following Microsoft documentation:

The documentation below is produced by a third-party vendor and thus is subject to change. MicroStrategy makes no guarantee on the availability or accuracy of third-party documentation.

- For Microsoft Windows 2003 *http://support.microsoft.com/kb/837361*

Configuring the krb5.ini file

If you use Intelligence Server Universal hosted on a Windows machine, you must configure the `krb5.ini` file. This file is included with an installation of MicroStrategy Web, and can be found in the following directory:

`C:\Program Files (x86)\Common Files\MicroStrategy\`

The path listed above assumes you have installed MicroStrategy in the `C:\Program Files (x86)` directory.

Kerberos only supports US-ASCII characters. Do not use any special characters when installing or configuring Kerberos.

Once you locate the `krb5.ini` file, open it in a text editor. The content within the file is shown below:

```
[libdefaults]
default_realm = <DOMAIN NAME>
default_keytab_name = <path to keytab file>
forwardable = true
no_addresses = true
[realms]
<REALM_NAME> = {
kdc = <IP address of KDC>:88
admin_server = <IP address of KDC admin>:749
}
[domain_realm]
.domain.com = <DOMAIN NAME>
domain.com = <DOMAIN NAME>
.subdomain.domain.com = <DOMAIN NAME>
subdomain.domain.com = <DOMAIN NAME>
```

You must configure the `krb5.ini` file to support your environment by replacing the entries enclosed in <>, which are described below:

- `<DOMAIN NAME>` and `<REALM_NAME>`: The domain realm used for authentication purposes. A domain realm is commonly of the form `EXAMPLE.COM`, and must be entered in uppercase.
- `<IP address of KDC>`: The IP address or host name of the Windows machine that hosts your Active Directory domain controller. This can be the same address as `<IP address of KDC admin>`.
- `<IP address of KDC admin>`: The host name or IP address of the Windows machine that hosts your Active Directory domain controller administration server. This can be the same address as `<IP address of KDC>`.
- `domain.com` and `subdomain.domain.com`: Use this for all domains and subdomains whose users must be authenticated using the default Kerberos realm.

Enabling integrated authentication for J2EE-compliant application servers

If you use a J2EE-compliant application server to deploy MicroStrategy Web Universal, MicroStrategy Mobile Server, or to deploy MicroStrategy Web Services to support MicroStrategy Office, you can support integrated authentication.

To enable integrated authentication, you must set up a Service Principal Name (SPN) for the application server, and configure the Kerberos keytab and configuration files. The following is an overview of the tasks you need to perform:

- *Create a Service Principal Name for your application server, page 151*

 © 2016, MicroStrategy Inc.

- *Configure the krb5.keytab file for the application server, page 151*
- *Enabling integrated authentication for J2EE-compliant application servers, page 150*
- *Configure the jaas.conf file for the application server, page 154*
- *Configure the JVM startup parameters, page 155*
- *Enable the SPNEGO mechanism, page 155*

Create a Service Principal Name for your application server

You must create a Service Principal Name (SPN) for your J2EE application server, and map it to the domain user that the application server runs as. The SPN identifies your application server as a service that uses Kerberos. For instructions on creating an SPN, refer to the Kerberos documentation.

The SPN should be in the following format:

`HTTP/ASMachineName`

The format is described below:

- `HTTP`: This is the service class for the application server.
- *`ASMachineName`*: This is the fully qualified host name of the server where the application server is running. It is of the form `machine-name.example.com`.

In your Active Directory, you must configure the application server's domain user to be trusted for delegation, and map the user to this SPN. For example, if your application server runs as the user `j2ee-http`, you must enable the **Account is trusted for delegation** option for the user, and map the user to the SPN. You must also enable the **Trust this computer for delegation to any service (Kerberos only)** option for the machine where your application server is hosted.

Configure the krb5.keytab file for the application server

You must create and configure a `krb5.keytab` file for the application server. In UNIX, you must use the `kutil` utility to create this file. In Windows, you must use the `ktpass` utility to create the keytab file.

The steps to configure this file on your UNIX or Linux machine are provided in *To create a krb5.keytab file in UNIX or Linux, page 152*.

The steps to configure this file on a Windows machine are provided in *To create a krb5.keytab file in Windows, page 153*.

Prerequisites

The procedure below requires a few variables to be entered for various commands. This includes information you can gather before you begin the procedure. The required variables in the following procedure are described below:

- *ASMachineName*: The name of the machine that the application server is installed on.
- *KeyVersionNumber*: The key version number, retrieved as part of this procedure.
- *DOMAIN_REALM*: The domain realm for the application server. It is of the form `EXAMPLE.COM`, and must be entered in uppercase.
- *EncryptionType*: The encryption type used.

It is recommended that you use `rc4-hmac` as the encryption type. Other encryption types may cause compatibility issues with the Windows Active Directory.

- *Keytab_Path*: For J2EE application servers under Windows, this specifies the location of the krb5.keytab file. It is of the form `C:\temp\example.keytab`.
- *ASUser* and *ASUserPassword*: The user account under which the application server runs, and the password for the account.

To create a krb5.keytab file in UNIX or Linux

If your application server and Intelligence Server are hosted on the same machine, it is required that you use separate keytab and configuration files for each. For example, if you are using `krb5.keytab` and `krb5.conf` for the Intelligence Server, use `krb5-http.keytab` and `krb5-http.conf` for the application server.

1. Log in to your UNIX or Linux machine.
2. Retrieve the key version number for your application server service principal name, using the command shown below:

   `kvno HTTP/`*`ASMachineName`*`@`*`DOMAIN_REALM`*

   The variables are described in the prerequisites above.

   The key version number is displayed on the command line.
3. In the command line, type the following commands:

If your application server is installed on the same machine as the Intelligence Server, replace `krb5.keytab` below with a different file name than the one used for the Intelligence Server, such as `krb5-http.keytab`.

```
ktutil

addent -password -p HTTP/ASMachineName@DOMAIN_NAME -k
KeyVersionNumber -e EncryptionType

wkt /etc/krb5/krb5.keytab

exit
```

 © 2016, MicroStrategy Inc.

4 To verify the keytab file, type the following command:

```
kinit -k -t /etc/krb5/krb5.keytab HTTP/ASMachineName
```

The command should run without prompting you for a password.

To create a krb5.keytab file in Windows

1 Log in to your Windows machine.

2 From a command prompt, type the following command:

```
ktpass -out Keytab_Path
-princ HTTP/ASMachine@DOMAIN_REALM
-mapUser ASUser
-mapOp set
-pass ASUserPassword
-crypto Encryption_Type
-pType KRB5_NT_PRINCIPAL
```

Configure the krb5.conf file for the application server

You must create and configure a file named `krb5.conf`.

For UNIX/Linux only: If your application server and Intelligence Server are hosted on the same machine, it is required that you use a separate configuration file. For example, if you created `krb5.conf` for the Intelligence Server, use `krb5-http.conf` for the application server.

If you have created a different keytab file in *Configure the krb5.keytab file for the application server, page 151*, replace `krb5.keytab` below with your own keytab file.

The contents of the `krb5.conf` should be as shown below:

```
[libdefaults]
default_realm = DOMAIN_REALM
default_keytab_name = Keytab_Path
forwardable = true
no_addresses = true

[realms]
DOMAIN_REALM = {
kdc = DC_Address:88
admin_server = DC_Admin_Address:749
}

[domain_realm]
.domain.com = DOMAIN_REALM
domain.com = DOMAIN_REALM
```

```
.subdomain.domain.com = DOMAIN_REALM
subdomain.domain.com = DOMAIN_REALM
```

The variables in the syntax above are described below:

- *DOMAIN_REALM*: The domain realm used for authentication purposes. A domain realm is commonly of the form `EXAMPLE.COM`, and must be entered in uppercase
- *Keytab_Path*: The location of your `krb5.keytab` file. In UNIX or Linux, it is of the form `/etc/krb5/krb5.keytab`. In Windows, it is of the form `C:\temp\krb5.keytab`.
- *domain.com* and *subdomain.domain.com*: Use this for all domains and subdomains whose users must be authenticated using the default Kerberos realm.
- *DC_Address*: The host name or IP address of the Windows machine that hosts your Active Directory domain controller. This can be the same address as *DC_Admin_Address*.
- *DC_Admin_Address*: The host name or IP address of the Windows machine that hosts your Active Directory domain controller administration server. This can be the same address as *DC_Address*.

Configure the jaas.conf file for the application server

You must configure the Java Authentication and Authorization Service (JAAS) configuration file for your application server.

Depending on the version of the Java Development Kit (JDK) used by your application server, the format of the `jaas.conf` file varies slightly. Refer to your JDK documentation for the appropriate format. Sample jaas.conf files for the Sun and IBM JDKs follow. The following variables are used:

- *ASMachineName*: The name of the machine that the application server is installed on.
- *DOMAIN_REALM*: The domain realm used for authentication purposes. It is of the form `EXAMPLE.COM`, and must be entered in uppercase.

Note that the parameters are entered in the `.accept` section of the `jaas.conf` file.

Sample jaas.conf for Sun JDK 1.5

```
com.sun.security.jgss.accept {
com.sun.security.auth.module.Krb5LoginModule required
principal="HTTP/ASMachineName@DOMAIN_REALM"
useKeyTab=true
doNotPrompt=true
storeKey=true
debug=true;
};
```

Sample jaas.conf for Sun JDK 1.6

 © 2016, MicroStrategy Inc.

```
com.sun.security.jgss.krb5.accept {
com.sun.security.auth.module.Krb5LoginModule required
principal="HTTP/ASMachineName@DOMAIN_REALM"
useKeyTab=true
doNotPrompt=true
storeKey=true
debug=true;
};
```

Sample jaas.conf for IBM JDK

```
com.ibm.security.jgss.accept {
com.ibm.security.auth.module.Krb5LoginModule required
useDefaultKeytab=true
principal="HTTP/ASMachineName@DOMAIN_REALM"
credsType=acceptor
forwardable=true
debug=true
storeKey=true;
};
```

Save the `jaas.conf` file to the same location as your `krb5.conf` file.

Configure the JVM startup parameters

For your J2EE-compliant application server, you must set the appropriate JVM startup parameters. The variables used are described below:

- *JAAS_Path*: The path to the `jaas.conf` file. In UNIX/Linux, it is of the form `/etc/krb5/jaas.conf`. In Windows, it is of the form `C:\temp\jaas.conf`.
- KRB5_Path: The path to the krb5.conf file. In UNIX/Linux, it is of the form `/etc/krb5/krb5.conf`. In Windows, it is of the form `C:\temp\krb5.conf`.

You must modify the JVM startup parameters listed below:

- `-Djava.security.auth.login.config=`*JAAS_Path*
- `-Djava.security.krb5.conf=`*KRB5_Path*
- `-Djavax.security.auth.useSubjectCredsOnly=false`

Enable the SPNEGO mechanism

As part of a MicroStrategy Web Universal or Mobile Server JSP deployment, you must modify the `web.xml` file for MicroStrategy Web or Mobile, to enable the Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO). This is accomplished by removing the comments around the following information in the `web.xml` file:

For MicroStrategy Web:

```
<filter>
```

```
      <display-name>SpnegoFilter</display-name>
      <filter-name>SpnegoFilter</filter-name>
      <filter-
      class>com.microstrategy.web.filter.SpnegoFilter</fil
      ter-class>
   </filter>
   <filter-mapping>
      <filter-name>SpnegoFilter</filter-name>
      <servlet-name>mstrWeb</servlet-name>
   </filter-mapping>
```

For MicroStrategy Mobile Server:

```
   <filter>
      <display-name>SpnegoFilter</display-name>
      <filter-name>SpnegoFilter</filter-name>
      <filter-
      class>com.microstrategy.mobile.filter.SpnegoFilter</
      filter-class>
   </filter>
   <filter-mapping>
      <filter-name>SpnegoFilter</filter-name>
      <servlet-name>mstrMobileAdmin</servlet-name>
</filter-mapping>
```

Restart your application server for all the above settings to take effect.

Configuring a browser for single sign-on to MicroStrategy Web

If a MicroStrategy Web user plans to use single sign-on to log in to MicroStrategy Web, the user must configure their browser to enable integrated authentication. The process to enable integrated authentication is different depending on the browser you use:

- For Internet Explorer, you must enable integrated authentication for the browser, as well as add the MicroStrategy Web server URL as a trusted site.

  Depending on your security policy, integrated authentication may be enabled by default for Internet Explorer.

- For Firefox, you must add the MicroStrategy Web server URL as a trusted site. The URL must be listed in the `about:config` page, in the settings `network.negotiate-auth.trusted-uris` and `network.negotiate-auth.delegation-uris`.

Linking a domain user to a MicroStrategy user

To apply security and privileges to a user in MicroStrategy, you must link the domain user to a MicroStrategy user. This also enables the domain user to be logged into

 © 2016, MicroStrategy Inc.

MicroStrategy projects they have access to without having to type their login credentials again.

Prerequisites

- A domain user included in a domain to support integrated authentication. For information on configuring a user in a domain, see *Configuring a domain controller and users, page 143*.
- A MicroStrategy user (object) to link to a domain user.
- A MicroStrategy user with administrative privileges to make the required user modifications.

To link a domain user to a MicroStrategy user

1. In Developer, log in to a project source using an account with administrative privileges.
2. From the **Folder List**, expand a project source, expand **Administration**, and then expand **User Manager**.
3. Browse to the MicroStrategy user you want to link a Windows user to. Right-click the MicroStrategy user and select **Edit**. The User Editor opens.
4. Expand **Authentication**, then select **Metadata**.
5. In the **Trusted Authenticated Request** area, type the domain user in the **User ID** field. Valid syntax is shown below:

   `DomainUserName@DOMAIN_REALM`

   For example, to link User1 who is in the `example.com` domain realm, you must type `User1@EXAMPLE.COM`. The domain realm name must be in uppercase.
6. Click **OK** to save your changes and close the User Editor.

Using integrated authentication for a project source

To enable users to log in to a project source in MicroStrategy with integrated authentication, you must define the project source to use integrated authentication. The procedure below describes the steps to define a project source to use integrated authentication.

Prerequisites

- A MicroStrategy user with administrative privileges to make the required user modifications.

© 2016, MicroStrategy Inc.

To use integrated authentication for a project source

1. In Developer, log in to a project source using an account with administrative privileges.
2. Right-click a project source, and then click **Modify Project Source**. The Project Source Manager opens.
3. On the **Connection** tab, under **Server Name**, type the server name exactly as it appears is the Service Principal Name created in *Trusting Intelligence Server for delegation, page 143*. For example, if the SPN is `MSTRSVRSvc\server.example.com:1234`, the Server Name for the project source should be `server.example.com`.
4. On the **Advanced** tab, select the **Use Integrated Authentication** option.

Enabling integrated authentication login mode for MicroStrategy Web

For MicroStrategy Web users to be able to use their Windows credentials to log in to MicroStrategy Web, you must enable integrated authentication as an available login mode. The procedure below describes the required steps for this configuration.

To enable integrated authentication login mode for MicroStrategy Web

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Web Administrator**. The Web Administrator page opens in a browser.
2. On the left, select **Default Properties**.
3. In the **Login** area, for **Integrated Authentication**, select the **Enabled** check box.

   If you want integrated authentication to be the default login mode for MicroStrategy Web, for **Integrated Authentication**, select the **Default** option.
4. Click **Save** to save your changes.

Enabling integrated authentication for MicroStrategy Mobile

To allow your MicroStrategy Mobile users to use their Windows credentials to log into MicroStrategy, you create a Mobile configuration, and select Integrated Authentication as the authentication method. For steps to create a Mobile configuration for your organization, see the *Mobile Server Help*.

 © 2016, MicroStrategy Inc.

Enabling integrated authentication to data sources

Through the use of integrated authentication, you can allow each user's credentials to be passed to your database server. You must enable this option at the project level.

If your reports or documents use subscriptions, using integrated authentication for your data sources prevents the subscriptions from running.

The steps to configure this optional support are described below.

Prerequisites

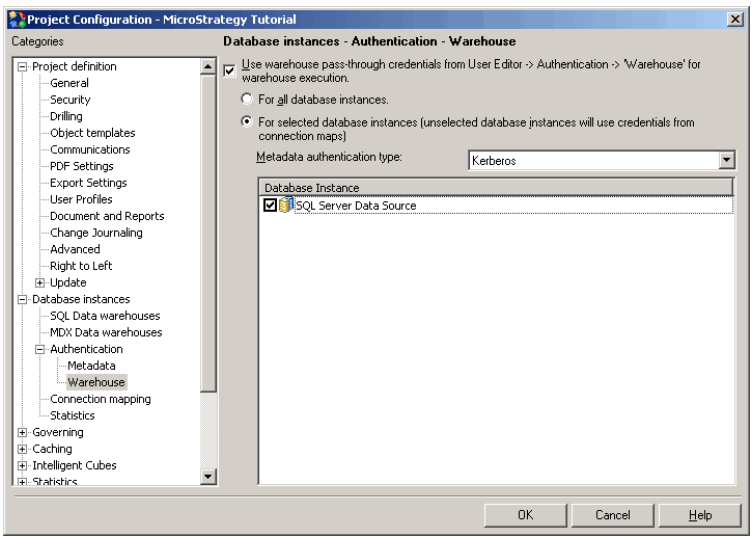
- Your database server must be configured to allow integrated authentication for all MicroStrategy users that use it as a data warehouse. Refer to your third-party database server documentation for instructions on enabling this support.

To enable integrated authentication to data sources

1. In Developer, log in to the project whose data sources you want to configure.
2. In the **Administration** menu, select **Projects**, then choose **Project Configuration**. The Project Configuration dialog box opens.
3. Expand the **Database instances** category.
4. Expand **Authentication**, and select **Warehouse**.
5. Enable the **For selected database instances** radio button.
6. From the **Metadata authentication type** drop-down list, choose **Kerberos**.
7. In the Database Instance pane, enable the check boxes for all the database instances for which you want to use integrated authentication, as shown below.

If you are connecting to a Microsoft SQL Server, Teradata, or TM1 data source, use this setting only if your Intelligence Server is running on Windows.

© 2016, MicroStrategy Inc.



8 Click **OK**. The Project Configuration dialog box closes.

Enabling integrated authentication for the MicroStrategy Hadoop Gateway

The MicroStrategy Hadoop Gateway is a data processing engine that you install in your Hadoop® environment. The Hadoop Gateway lets you analyze unstructured data in Hadoop, and provides high-speed parallel data transfer between the Hadoop Distributed File System (HDFS) and your MicroStrategy Intelligence Server.

Before enabling integrated authentication for your Hadoop cluster, ensure that you have met the following prerequisites. To enable integrated authentication for your Hadoop cluster, refer to your third-party documentation.

Prerequisites

- You have installed the Hadoop Gateway in your Hadoop cluster. For steps to install the Hadoop Gateway, see the Installation and Configuration Guide.
- You have enabled integrated authentication for Intelligence Server and Web, as described in *Enabling integrated authentication for IIS, page 147* or *Enabling integrated authentication for J2EE-compliant application servers, page 150*, depending on your platform.
- In your Hadoop cluster, you have set up a user account that has permissions to create new users.
- Your Hadoop cluster can access your Kerberos domain controller.

For specific steps to enable integrated authentication for your Hadoop cluster, refer to the documentation for your Hadoop cluster distribution.

 © 2016, MicroStrategy Inc.

Enabling Single Sign-on to Web, Mobile, and Office with third-party authentication

You can enable Single Sign-on (SSO) authentication for the following MicroStrategy applications using a third-party tool such as IBM Tivoli Access Manager, CA SiteMinder, Oracle Access Manager, or PingFederate®:

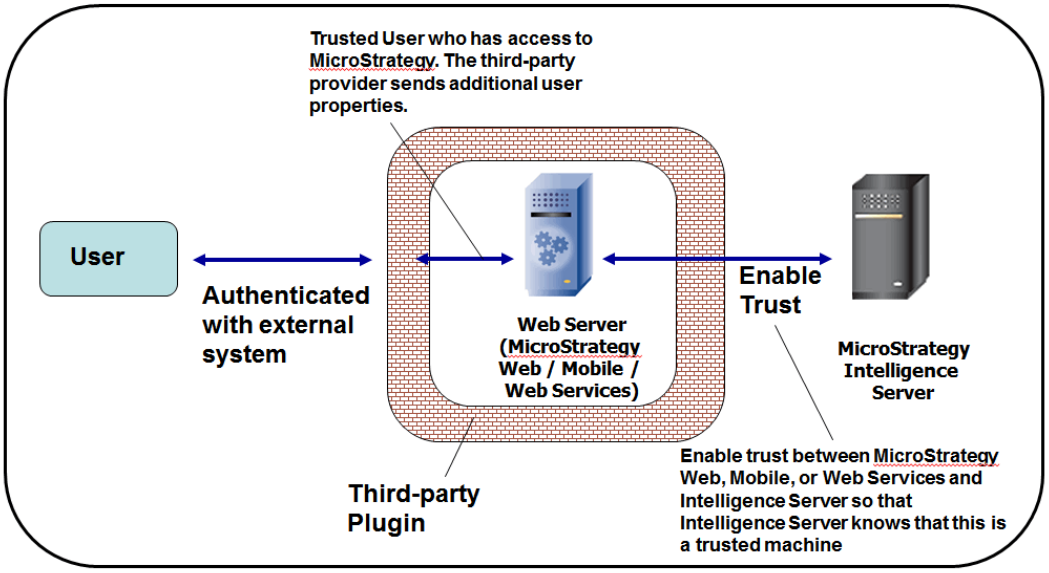
- MicroStrategy Web
- MicroStrategy Mobile
- MicroStrategy Web Services, to support MicroStrategy Office (IBM Tivoli Access Manager and CA SiteMinder only)

Once a user is authenticated in the third-party system, the user's permissions are retrieved from a user directory, such as LDAP, and access is granted to the MicroStrategy application.

In this security model, there are several layers. For example, when a user logs in to Tivoli, Tivoli determines whether or not the user's credentials are valid. If the user logs in with valid credentials to Tivoli, the user directory (such as LDAP) determines whether that valid user can connect to MicroStrategy. The user's MicroStrategy privileges are stored within the MicroStrategy Access Control List (ACL). What a user can and cannot do within the MicroStrategy application is stored on Intelligence Server in the metadata within these ACLs. For more information about privileges and ACLs in MicroStrategy, see *Chapter 2, Setting Up User Security*.

For MicroStrategy to be able to get a user's privileges from the metadata, Intelligence Server must be configured to be a trusted machine in MicroStrategy Web, Mobile, and Office. This allows the information to be passed between the two machines.

The following diagram illustrates the architecture of a security system that uses third-party authentication.



MicroStrategy enables this type of access by passing tokens between MicroStrategy, the user directory, and the third-party authentication provider. Properly configuring these levels of communication is critical to implementing SSO authentication.

The distinguished name of the user passed from the third-party provider is URL-decoded by default within MicroStrategy Web, Mobile, or Web Services before it is passed to the Intelligence Server.

Single sign-on authentication performs the step of allowing a user access to MicroStrategy products. You also must configure MicroStrategy users to define privileges and permissions that control what a user can perform and access within the products.

Setting up third-party SSO authentication in MicroStrategy products

The following high-level steps are required to set up third-party SSO authentication in MicroStrategy Web, Mobile, or Web Services, and each is detailed below:

- *Creating users and links in third-party authentication systems, page 162*
- *Enabling single sign-on authentication to MicroStrategy Web, Mobile or Office, page 163*
- *Importing and linking third-party authentication users in MicroStrategy, page 172*
- *To log in to MicroStrategy Web using Tivoli single sign-on, page 176*

Creating users and links in third-party authentication systems

Before MicroStrategy can be configured to accept Tivoli, SiteMinder, PingFederate or Oracle Access Manager authentication, certain preliminary settings must be established. This ensures that a link exists between the authentication provider and MicroStrategy products, and that the link is functioning as required.

You must complete all of the following steps to ensure proper configuration of your authentication provider and MicroStrategy products.

Creating a user in your third-party authentication system

You can enable SSO authentication in MicroStrategy by associating a MicroStrategy user to a user in Tivoli, SiteMinder, PingFederate or Oracle Access Manager. To test this association, you must create a user in your authentication system to confirm that access has been properly configured in MicroStrategy products.

For steps to create a new user, refer to your authentication provider's documentation.

Creating a link to MicroStrategy applications in your third-party authentication system

You link Tivoli to MicroStrategy applications using junctions, SiteMinder using Web Agents, and Oracle Access Manager using Webgates. These links redirect users from the

© 2016, MicroStrategy Inc.

respective provider to MicroStrategy, and are required to enable SSO authentication. You must create one link each, as applicable, for MicroStrategy Web, MicroStrategy Mobile, and MicroStrategy Web Services to support MicroStrategy Office.

Oracle Access Manager authentication is only available for MicroStrategy Web.

For steps to create a junction (in Tivoli), a Web Agent (in SiteMinder), or a Webgate (Oracle Access Manager), refer to the product's documentation.

Enabling single sign-on authentication to MicroStrategy Web, Mobile or Office

Once the initial third-party authentication setup is complete, you must enable trusted authentication in MicroStrategy Web, Mobile or Office, and establish trust between the MicroStrategy product and Intelligence Server. This allows the authentication token to be passed from one system to the other.

Note that for MicroStrategy Web Services to support MicroStrategy Office, you must establish trust between Office and the Intelligence server, and enable trusted authentication in the configuration files for Web Services.

This section explains the following required steps to enable SSO authentication in MicroStrategy Web, Mobile, or Web Services:

- *Enabling trusted authentication in MicroStrategy Web, page 163*
- *Enabling trusted authentication in MicroStrategy Mobile, page 165*
- *Establishing trust between MicroStrategy Web or Mobile and Intelligence Server, page 165*
- *Establishing trust between MicroStrategy Web Services and Intelligence Server, to support MicroStrategy Office, page 168*
- *Enabling trusted authentication in MicroStrategy Web Services to support MicroStrategy Office, page 169*

If you use Internet Information Services (IIS) as your web server for MicroStrategy Web or Web Services, you must enable anonymous authentication to the MicroStrategy virtual directories to support SSO authentication to MicroStrategy Web, Mobile, or Office. This is discussed in *Enabling anonymous authentication for Internet Information Services, page 171*.

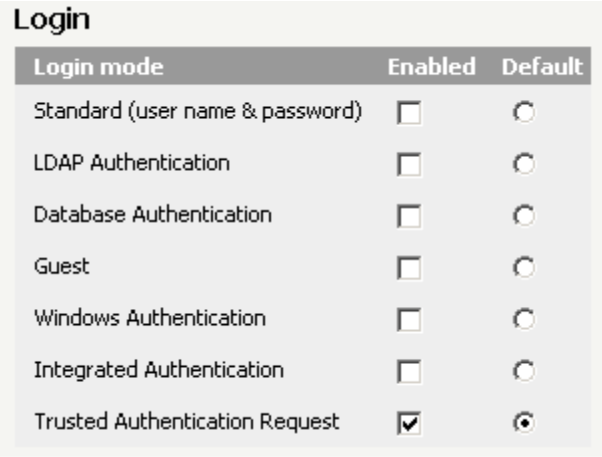
Enabling trusted authentication in MicroStrategy Web

To enable users to log in to MicroStrategy Web using SSO authentication, you must enable trusted authentication as an available authentication mode in MicroStrategy Web.

© 2016, MicroStrategy Inc.

To enable trusted authentication in MicroStrategy Web

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Web Administrator**. The MicroStrategy Web Administrator page opens.
2. On the left side of the page, click **Default Properties**. The Default Properties page opens.
3. Scroll down to the Login area and, under **Login mode**, select the **Enabled** check box next to **Trusted Authentication Request**. Also select the **Default** option next to **Trusted Authentication Request,** as shown below:



4. From the **Trusted Authentication Providers** drop-down list, select **IBM Tivoli Access Manager**, **CA SiteMinder**, **PingFederate**, or **Oracle Access Manager**.

   To use a custom authentication provider, select **Custom SSO**. For information about adding custom authentication providers, refer to your MicroStrategy SDK documentation.
5. At the bottom of the page, click **Save**.

Using certificate authentication with SiteMinder

CA SiteMinder can be configured to use either certificate authentication or basic authentication. MicroStrategy Web's `siteminder_security.properties` file indicates that the first SiteMinder header variable to be used is `SM_UNIVERSALID`. This variable provides information for certificate authentication. If this variable is empty, then the information in the variable `SM_USER` is used for basic authentication. For information about configuring your SiteMinder system to use certificate authentication, see the SiteMinder documentation.

© 2016, MicroStrategy Inc.

Enabling trusted authentication in MicroStrategy Mobile

To enable users to log in to MicroStrategy Mobile using SSO authentication, you must enable trusted authentication as an available authentication mode in MicroStrategy Mobile. For instructions on configuring mobile devices to use trusted authentication, refer to the *Administering MicroStrategy Mobile* chapter in the MicroStrategy Mobile Design and Administration Guide.

To enable trusted authentication in MicroStrategy Mobile

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Mobile Administrator**. The MicroStrategy Mobile Server Administrator page opens.
2. On the left side of the page, click **Default Properties**. The Default Properties page opens.
3. From the **Trusted Authentication Providers** drop-down list, select **IBM Tivoli Access Manager**, **CA SiteMinder**, **PingFederate**, or **Oracle Access Manager**.

   To use a custom authentication provider, select **Custom SSO**. For information about adding custom authentication providers, refer to your MicroStrategy SDK documentation.
4. At the bottom of the page, click **Save**.

To create a mobile configuration to send to users' mobile devices, refer to the *Administering MicroStrategy Mobile* chapter in the MicroStrategy Mobile Design and Administration Guide.

Establishing trust between MicroStrategy Web or Mobile and Intelligence Server

To enable the authentication token to pass from your third-party authentication provider to MicroStrategy Web or Mobile, and then to Intelligence Server, a trust relationship must be established between MicroStrategy Web or Mobile and Intelligence Server. The steps to establish trust are described below.

If you need to delete an established trust relationship, see *To delete a trust relationship, page 167*.

If you are using multiple Intelligence Server machines in a cluster, you must first set up the cluster, as described in *Chapter 9, Clustering Multiple MicroStrategy Servers*, and then establish trust between Web or Mobile Server and the cluster.

Prerequisites

To establish trust between MicroStrategy Web or Mobile and Intelligence Server, you must have the following privileges:

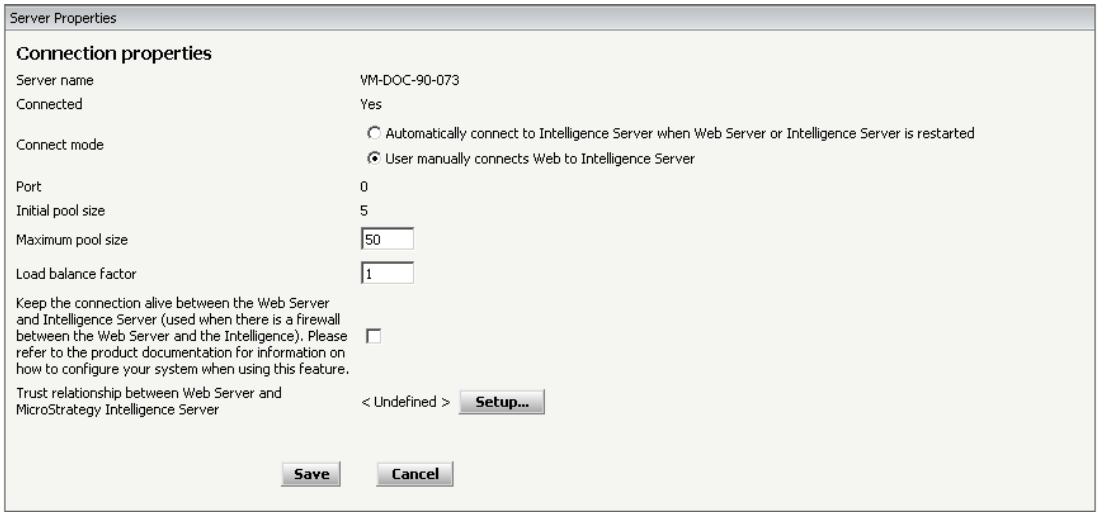
- Bypass all object security access checks

- Configure security settings
- Enable Intelligence Server administration from Web
- Web administration

For information on assigning privileges to users, see *Chapter , Controlling access to functionality: Privileges*.

To establish trust between MicroStrategy Web or Mobile and Intelligence Server

1. Open MicroStrategy Web Administrator or MicroStrategy Mobile Administrator, as applicable:
   - From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Web Administrator**. The MicroStrategy Web Administrator page opens.
   - From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Mobile Administrator**. The MicroStrategy Mobile Server Administrator page opens.
2. On the left, click **Servers**.
3. Confirm that MicroStrategy Web or Mobile Server is currently connected to an Intelligence Server. If an Intelligence Server is not connected, in the **Unconnected Servers** table, under **Action**, click **Connect** for the appropriate Intelligence Server.
4. In the **Connected Servers** table, under **Properties**, click the **Modify** icon . The Server Properties page opens, as shown below.



5. Next to **Trust relationship between** *Web/Mobile* **Server and MicroStrategy Intelligence Server**, as applicable, click **Setup**. The Setup Trust Relationship with MicroStrategy Intelligence Server page opens.

© 2016, MicroStrategy Inc.

6 Type a **User name** and **Password** in the appropriate fields. The user must have administrative privileges for MicroStrategy Web or Mobile, as applicable.

7 From the options provided, select the authentication mode used to authenticate the administrative user.

8 In the **Web Server Application** or **Mobile Server Application** field, type a unique name for the trust relationship.

For example, you can use the URLs for the applications using Tivoli, as follows:

MicroStrategy Web:
`https://MachineName/JunctionName/MicroStrategy/asp`

MicroStrategy Mobile:
`https://MachineName/JunctionName/MicroStrategyMobile/asp`

9 Click **Create Trust Relationship**. The Server Properties page opens.

10 Click **Save** to create the trust relationship.

To verify the trust relationship

11 From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **Developer**. Developer opens.

12 Log in to a project source as a user with administrative privileges.

13 From the **Administration** menu, point to **Server**, and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.

14 On the left, expand the **Web Single Sign-on** category, and verify that the trusted relationship is listed in the **Trusted Web Application Registration** list.

15 Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.

To delete a trust relationship

1 Open MicroStrategy Web Administrator or MicroStrategy Mobile Administrator, as applicable:

- From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Web Administrator**. The MicroStrategy Web Administrator page opens.
- From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Mobile Administrator**. The MicroStrategy Mobile Server Administrator page opens.

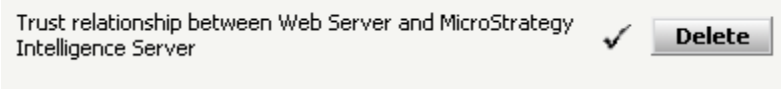
2 On the left, click **Servers**.

© 2016, MicroStrategy Inc.

3 Confirm that MicroStrategy Mobile is currently connected to an Intelligence Server. If an Intelligence Server is not connected, in the **Unconnected Servers** table, under **Action**, click **Connect** for the appropriate Intelligence Server.

4 In the **Connected Servers** table, under **Properties**, click the **Modify** icon . The Server Properties page opens.

5 Next to **Trust relationship between MicroStrategy** *Web/Mobile* **Server and MicroStrategy Intelligence Server**, as applicable, click **Delete**.



The Delete Trust Relationship with MicroStrategy Intelligence Server page opens.

6 Provide your login information in the appropriate fields.

7 Click **Delete trust relationship**.

The Server Properties page opens.

8 Click **Save**.

Establishing trust between MicroStrategy Web Services and Intelligence Server, to support MicroStrategy Office

To establish trust between MicroStrategy Office and Intelligence Server, you must use MicroStrategy Office to connect to the project source you want to use trusted authentication for, and then establish the trust relationship between Office and the

© 2016, MicroStrategy Inc.

Intelligence Server. Once you have completed this step, you must edit the `projectsources.xml` file for Web Services to enable trusted authentication for the project source. Both procedures are described below.

To establish trust between MicroStrategy Web Services and Intelligence Server

1. On a machine where MicroStrategy Office is installed, open a Microsoft Office product, such as Excel.
2. In the Microsoft Office ribbon, under the MicroStrategy Office tab, click **MicroStrategy Office**. MicroStrategy Office starts, with a list of project sources you can connect to.
3. From the list of project sources on the left, select the project source you want to enable trusted authentication for.
4. In the right pane, enter the login ID and password for a user with administrative privileges, and click **Get Projects**. A list of projects is displayed.
5. Select any project, and click **OK**. The project is loaded, and the Shared Reports folder is loaded in a new window.
6. In the MicroStrategy Office toolbar, click **Options**. The MicroStrategy Office Options dialog box opens.
7. Under the **General** category, select **Server**. The Server Options pane is displayed on the right.
8. Next to **Trust relationship between Web Services and Intelligence Server**, click **Create**. An alert is displayed when the trust relationship has been established.

To use the third-party authentication URL for Web Services

9. In the **Web Services URL** field, enter the URL for the Tivoli Junction or SiteMinder Web Agent, as applicable, that you created for MicroStrategy Web Services.
10. Click **OK** to save the changes.

Enabling trusted authentication in MicroStrategy Web Services to support MicroStrategy Office

To allow users to log in to MicroStrategy Office using single sign-on (SSO), you must do the following:

- Edit the `web.config` file for Web Services or `MWSConfig.properties` file for J2EE application servers, to choose a trusted authentication provider.
- Edit the `projectsources.xml` file for MicroStrategy Web Services and configure the project source to use a third-party security plug-in. For additional information on

© 2016, MicroStrategy Inc.

the settings in the `projectsources.xml` file, refer to the *Installing and Administering MicroStrategy Office* chapter in the MicroStrategy Office User Guide.

Prerequisite

- You need administrative access to the machine where MicroStrategy Web Services is installed.

To enable trusted authentication in MicroStrategy Office

To choose a trusted authentication provider

1. Depending on your Web Services environment, on the machine where MicroStrategy Web Services is installed, do one of the following:
   - If you are using IIS as your application server, open the `web.config` file in a text editor, such as Notepad. By default, the file is located in `C:\Program Files (x86)\MicroStrategy\Web Services`.
   - If you are using Web Services in a J2EE-compliant application server, open the `MWSConfig.properties` file in a text editor, such as Notepad. By default, the file is located in the folder where your application server deploys Web Services.
2. Depending on your Web Services environment, do the following:
   - In the `web.config` file, locate the following line:

     ```
     <add key="TRUSTEDAUTHPROVIDER" value="1" />
     ```

   - In the `MWSConfig.properties` file, locate the following line:

     ```
     TRUSTEDAUTHPROVIDER=1
     ```

3. Change `value` or `TRUSTEDAUTHPROVIDER`, as applicable, to one of the following, as applicable:
   - To use Tivoli as the authentication provider, type `1`.
   - To use SiteMinder as the authentication provider, type `2`.
   - To use a custom authentication provider, type `3`.

If you are using a custom authentication provider, you must make additional modifications to the `custom_security.properties` file, which is located by default in `C:\Program Files (x86)\MicroStrategy\Web Services\resources`. For information on these modifications, refer to the *MicroStrategy Developer Library (MSDL)*.

© 2016, MicroStrategy Inc.

To configure Web Services to use trusted authentication

4 On the machine where MicroStrategy Web Services is installed, open the `projectsources.xml` file in a text editor, such as Notepad. By default, the file is located in `C:\Program Files (x86)\MicroStrategy\Web Services`.

5 In the `projectsources.xml` file, locate the `<ProjectSource>` tag describing the project source you want to enable SSO for.

6 In the `<ProjectSource>` tag, replace the content of the `<AuthMode>` tag with `MWSSimpleSecurityPlugin`. The contents of the new `<ProjectSource>` tag should appear similar to the following:

```
<ProjectSource>
<ProjectSourceName>Name</ProjectSourceName>
<ServerName>Name</ServerName>
<AuthMode>MWSSimpleSecurityPlugIn</AuthMode>
<PortNumber>0</PortNumber>
</ProjectSource>
```

7 Save `projectsources.xml`.

Enabling anonymous authentication for Internet Information Services

If you use Internet Information Services (IIS) as your web server, you must enable anonymous authentication to the MicroStrategy virtual directory to support SSO authentication to MicroStrategy Web, Web Services or Mobile.

The steps to perform this configuration are provided below, which may vary depending on your version of IIS. The following links may contain information to enable anonymous authentication for your version of IIS:

- **IIS 7**: *http://technet.microsoft.com/en-us/library/cc770966(WS.10).aspx*
- **IIS 6**: *http://technet.microsoft.com/en-us/library/cc737887(WS.10).aspx*
- **IIS 5**: *http://support.microsoft.com/kb/310344*

You cannot use Windows authentication to authenticate users in MicroStrategy Web or Mobile if you enable anonymous authentication to the MicroStrategy virtual directory in IIS. The steps below should only be used as part of an SSO authentication solution with Tivoli.

To configure IIS to enable anonymous authentication to the MicroStrategy Web, Web Services, and Mobile virtual directories

1 On the MicroStrategy Web server machine, access the IIS Internet Service Manager.

2 Browse to and right-click the **MicroStrategy** virtual folder and select **Properties**.

3 On the **Directory Security** tab, under **Anonymous access and authentication control**, click **Edit**. The Authentication Methods dialog box opens

4 Select the **Allow anonymous access** check box.

5 Click **OK** to save your changes and close the Authentication Methods dialog box.

6 Click **OK** again to save your changes to the MicroStrategy virtual folder.

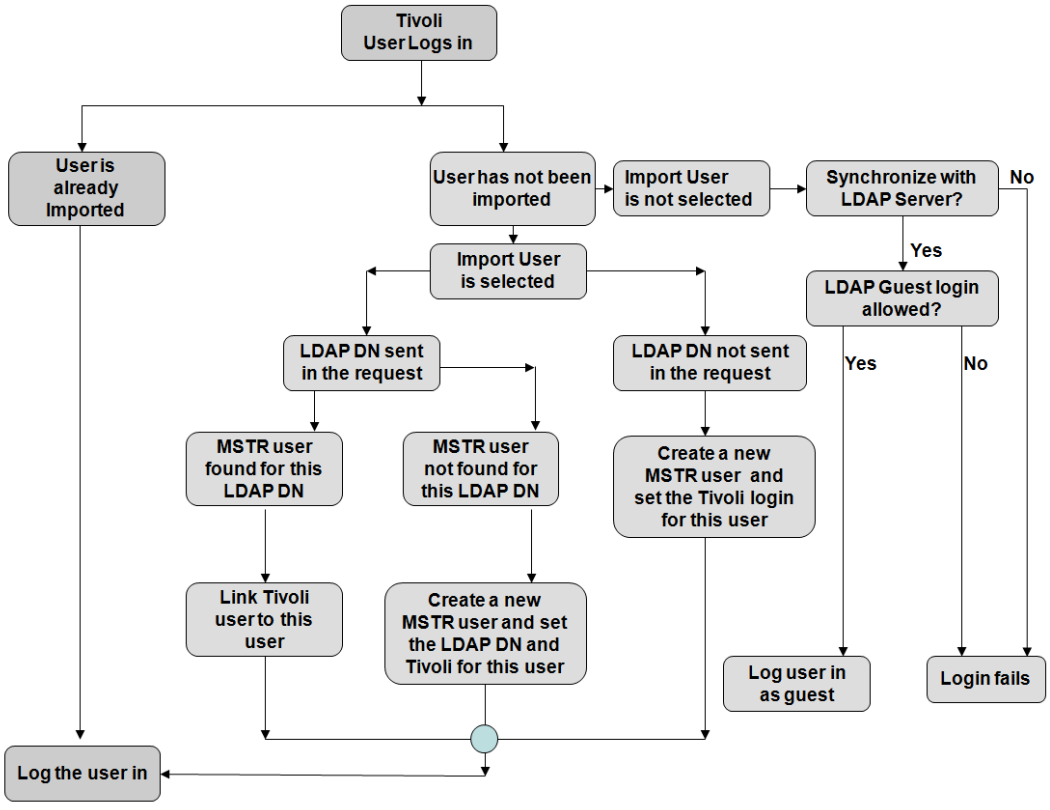
7 To enable anonymous authentication to MicroStrategy Web Services, repeat the above procedure for the **MicroStrategyWS** virtual directory.

8 To enable anonymous authentication to MicroStrategy Mobile, repeat the above procedure for the **MicroStrategyMobile** virtual directory on the Mobile Server machine.

9 Restart IIS for the changes to take effect.

Importing and linking third-party authentication users in MicroStrategy

For third-party authentication users to access MicroStrategy applications, the users must be granted MicroStrategy privileges. The following flowchart illustrates the various ways that MicroStrategy users are handled when they log in to Tivoli or SiteMinder:



© 2016, MicroStrategy Inc.

Whether or not the LDAP DN is sent in the request to Intelligence Server is configured when the Tivoli junction or SiteMinder Web Agent is created. For details about creating a junction or Web Agent, refer to your Tivoli or SiteMinder documentation.

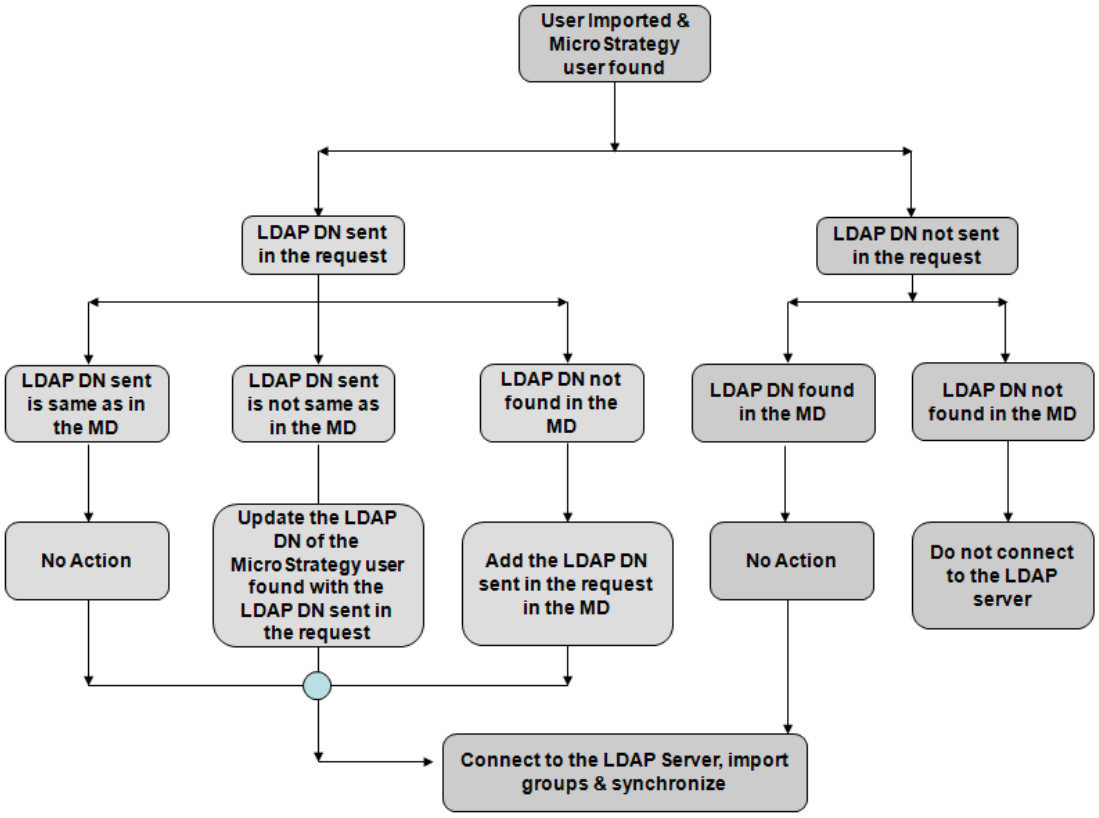
A Tivoli or SiteMinder user can be:

- Imported as a new MicroStrategy user upon logging in to MicroStrategy Web, which assigns the user privileges that are defined for the MicroStrategy user. For steps to perform this configuration, see *Importing Tivoli users as MicroStrategy users, page 174*.
- Allowed guest access to MicroStrategy Web. The Tivoli user inherits the privileges of the Public/Guest group in MicroStrategy. Guest access to MicroStrategy Web is not necessary for imported or linked Tivoli users. For steps to perform this configuration, see *Enabling guest access to MicroStrategy Web or Mobile for Tivoli users, page 175*.

A Tivoli or SiteMinder user can also be associated with an existing MicroStrategy user, using the MicroStrategy User Editor. Associating Tivoli users rather than enabling Tivoli users to be imported when they log in to MicroStrategy Web enables you to assign MicroStrategy privileges and other security settings for the user prior to their initial login. For steps to perform this configuration, see *Linking Tivoli users to existing MicroStrategy users, page 175*.

If a Tivoli or SiteMinder user has already been imported into MicroStrategy, and a MicroStrategy user has been associated with the Tivoli or SiteMinder user, the MicroStrategy metadata is synchronized with the information from the user directory, such as the LDAP server. The way this synchronization takes place depends upon several factors. The following flowchart shows how the synchronization is handled.

© 2016, MicroStrategy Inc.



Importing Tivoli users as MicroStrategy users

When MicroStrategy is configured to import a Tivoli user, the Tivoli user is imported as a MicroStrategy user the first time that the user logs in to MicroStrategy Web after the configuration is completed. A Tivoli user is imported into MicroStrategy only if the Tivoli user has not already been imported as or associated with a MicroStrategy user.

When a Tivoli user is imported into MicroStrategy:

- The Tivoli user name is imported as the trusted authentication request user ID for the new MicroStrategy user.
- The MicroStrategy user is added to the Everyone group by default. If no privileges are defined through a user directory such as LDAP, then the imported user inherits the privileges associated with the MicroStrategy Everyone group.
- Security privileges are not imported from Tivoli; these must be defined in MicroStrategy by an administrator.

To import Tivoli users as MicroStrategy users

1 From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **Developer**. Developer opens.

© 2016, MicroStrategy Inc.

2 Log in to a project source as a user with administrative privileges.

3 From the **Administration** menu, point to **Server**, and then **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.

4 On the left, expand the **Web Single Sign-on** category.

5 On the right, select the **Import user at login** check box.

6 Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.

Linking Tivoli users to existing MicroStrategy users

As an alternative to importing users, you can link (or associate) Tivoli users to existing MicroStrategy users to retain the existing privileges and configurations defined for the MicroStrategy users. Linking Tivoli users rather than enabling Tivoli users to be imported when they log in to MicroStrategy Web enables you to assign privileges and other security settings for the user prior to their initial login.

To link Tivoli users to existing MicroStrategy users

1 From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **Developer**. Developer opens.

2 Log in to a project source as a user with administrative privileges.

3 In the folder list on the left, expand **Administration,** and then expand **User Manager.**

4 Browse to the MicroStrategy user to link to a Tivoli user.

5 Right click the user and select **Edit**. The **User Editor** opens.

6 Expand **Authentication**, then select **Metadata**.

7 Under **Trusted Authentication Request**, in the **User ID** field, type the Tivoli user name to link to the MicroStrategy user.

The name you type in the **User ID** field should be the same as the one that the user employs when providing his Tivoli login credentials.

8 Click **OK**.

Enabling guest access to MicroStrategy Web or Mobile for Tivoli users

If you choose to not import or link Tivoli users to a MicroStrategy user, you can enable guest access to MicroStrategy Web for the Tivoli users. Guest users inherit their privileges from the MicroStrategy Public/Guest group.

© 2016, MicroStrategy Inc.

To enable guest access to MicroStrategy Web or Mobile for Tivoli users

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **Developer**. Developer opens.
2. Log in to a project source as a user with administrative privileges.
3. From the **Administration** menu, point to **Server**, and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
4. On the left, expand the **Web Single Sign-on** category.
5. On the right, in the **Trusted Web Application Registration** area, select the MicroStrategy Web or MicroStrategy Mobile application, as applicable.
6. Select the **Allow user to log in if Web Single Sign-on - MicroStrategy user link not found** check box.
7. On the right, expand the **LDAP** category, then expand the **Import** subcategory, and select **Options**.
8. Select the **Synchronize user/group information with LDAP during Trusted authentication** check box.
9. Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.

Logging in to MicroStrategy Web using Tivoli single sign-on

Once all of the preliminary steps have been completed and tested, users may begin to sign in to MicroStrategy using their Tivoli credentials. Sign-on steps are provided in the procedure below.

To log in to MicroStrategy Web using Tivoli single sign-on

1. Open a web browser.
2. Type the following URL in the address field:
   `https://MachineName/JunctionName/MicroStrategyWebURL`

   Where the variables in italics are as follows:

   - *`MachineName`* is the name of the machine running Tivoli.
   - *`JunctionName`* is the name of the junction created in Tivoli.
   - *`MicroStrategyWebURL`* is the URL to access MicroStrategy Web. For example, `MicroStrategy/asp`.
3. Type your Tivoli user name and password.

© 2016, MicroStrategy Inc.

4 Connect to a MicroStrategy project.

5 Click **Trusted Authentication**.

You are logged in to the MicroStrategy project with your Tivoli user credentials.

If you are prompted to display both secure and non-secure items on the web page, you can configure your web browser to hide this warning message. Refer to your web browser documentation regarding this configuration.

Enabling Usher authentication for Web and Mobile

If you use an LDAP directory to centrally manage users in your environment, you can add them to your Usher network, and allow them to log into MicroStrategy Web or Mobile by using their Usher badges.

The users in your LDAP directory can log into MicroStrategy Web by:

- Scanning a QR code using the Usher app on their smart phones, if Usher is configured as the primary authentication method.
- Supplementing their user name and password with a numerical Usher code that is provided via the Usher app on their smart phones, if Usher is configured as the second factor of authentication.

The high-level steps to enable Usher authentication for Web and Mobile are as follows:

1 Set up an Usher network, and badges for your users. Your network is the group of users in your organization who can use the Usher app on their smart phone to validate their identity to log in to MicroStrategy. For steps to create an Usher network, see the *Usher Help*.

2 Add your LDAP directory to your Usher network. For steps to add your LDAP directory to Usher, see the *Usher Help*.

3 Connect your LDAP directory to MicroStrategy. For steps to connect your LDAP directory to MicroStrategy, see *Implementing LDAP authentication, page 105*.

4 Register your MicroStrategy environment with Usher. For steps, see *Registering your MicroStrategy products with Usher, page 177*.

5 Configure Usher in MicroStrategy Web and Mobile. For steps, see *Configuring Usher in MicroStrategy Web and Mobile, page 178*.

Registering your MicroStrategy products with Usher

To establish a connection between Usher and your MicroStrategy products, you must register your MicroStrategy products with Usher, using the steps below.

Prerequisites

- You have created an Usher network, and badges for your users. Your network is the group of users in your organization who can use the Usher app on their smart phone

to validate their identity to log in to MicroStrategy. For steps to create an Usher network, see the *Usher Help*.

- You have connected an LDAP user directory to MicroStrategy. For steps to connect your LDAP directory to MicroStrategy, see *Implementing LDAP authentication, page 105*.

To register MicroStrategy with Usher

1. In a web browser, log in to Usher Network Manager, using one of the following methods:
   - If you have installed Usher in your IT infrastructure, navigate to the server where you installed Usher Network Manager.
   - If you are using Usher in the cloud, navigate to *https://go.usher.com*.
2. Click **Gateway Configuration**. The Gateway Configuration page opens.
3. In the MicroStrategy Platform Login area, click the **MicroStrategy** icon. Review the message, then click **Continue**.
4. To change the image that is displayed on the login page when users open MicroStrategy Web, click **Import an Icon**. Select an image to display, and click **Open**.
5. In the **Enter Display Name** field, type a name to display on your MicroStrategy login page.
6. Click **Next**. The Set Up Your MicroStrategy Platform page is shown, with the details to configure your MicroStrategy Intelligence Server.
7. Note the values for Organization ID, Application ID, and Token. You use these values to configure MicroStrategy Intelligence Server.
8. Click **Done**.

Configuring Usher in MicroStrategy Web and Mobile

To allow your users to log into MicroStrategy Web and Mobile with their Usher badges, you must configure Usher as a trusted authentication provider in Web Administrator and Mobile Administrator, as described in the steps below.

Prerequisites

- You have registered your MicroStrategy products with Usher, as described in *Registering your MicroStrategy products with Usher, page 177*, and noted the Organization ID, Application ID, and Token provided by Usher.
- You have upgraded your MicroStrategy metadata. For steps to upgrade your MicroStrategy metadata, see the Upgrade Guide.

 © 2016, MicroStrategy Inc.

Enabling Usher authentication without upgrading your metadata may cause your users to be locked out of MicroStrategy applications.

- If you are enabling two-factor authentication for Web using Usher, you have added at least one user to the Two-factor Exempt (2FAX) user group in your MicroStrategy project. MicroStrategy users who are members of the Two-factor Exempt (2FAX) group are exempt from two-factor authentication, and do not need to provide an Usher code to log into MicroStrategy Web. It is recommended that these users have a secure password for their accounts, and use their accounts for troubleshooting MicroStrategy Web.

Ensure that you configure your LDAP server information correctly in your Intelligence Server. If it is not configured correctly, two-factor authentication cannot be used and therefore users will not be able to log into the server.

To enable Usher authentication in Web and Mobile

To configure Intelligence Server for Usher authentication

1. From the Windows **Start** menu, select **All Programs**, then **MicroStrategy Tools**, and click **Web Administrator**. The Web Administrator page opens, with a list of your Intelligence Servers.
2. For your Intelligence Server, click **Modify**. The Server Properties page opens.
3. Next to Usher Configuration, click **Setup**. The Usher Configuration page opens.
4. In the Connectivity section, in the **Usher Server URL** field, enter one of the following:
   - If you are using Usher in the cloud, type `https://uvs-v2.usher.com`.
   - If you have installed Usher in your infrastructure, type the URL for the Usher Server you installed.
5. In the **OrgID** field, type the Organization ID from Usher Network Manager.
6. In the AppID field, type the **Application ID** from Usher Network Manager.
7. If you want to use Usher as a two-factor authentication system, select the **Enable two-factor authentication** check box. The Security token field is enabled.

   MicroStrategy users who are members of the Two-factor Exempt (2FAX) group are exempt from two-factor authentication, and do not need to provide an Usher code to log into MicroStrategy Web. It is recommended that these users have a secure password for their accounts, and use their accounts for troubleshooting MicroStrategy Web.
8. In the **Security token** field, type the Security Token from Usher Network Manager.

9 Click **Save** to save your changes. To save the Usher configuration, you are prompted for your MicroStrategy credentials.

10 Type your MicroStrategy user name and password, and click **Login**. The settings are saved.

To enable Usher authentication in Web and Mobile

11 In Web Administrator, from the menu on the left, click **Default Properties**. The Default Server Properties page opens.

12 In the Login area, for **Trusted Authentication Request**, select the **Enabled** check box.

13 From the **Trusted Authentication Providers** drop-down menu, select **Usher**.

14 Click **Save** to save your changes.

15 Repeat the above steps to add Usher as a trusted authentication provider in Mobile Administrator. To open Mobile Administrator, in the Start menu, select **All Programs**, then **MicroStrategy Tools**, and click **Mobile Administrator**.

Authentication examples

Below are a few examples of how the different methods for user authentication can be combined with different methods for database authentication to achieve the security requirements of your MicroStrategy system. These examples illustrate a few possibilities; other combinations are possible.

Security views: Windows authentication and linked warehouse login

You may want to use this configuration if you are using security views to implement access control policies for data. For example, two different users executing the same SQL query receive different results, reflecting their different levels of access. For the security views to work, each report is executed under the RDBMS account of the user who submitted the report from the MicroStrategy system. Even though this approach requires users to have accounts on the RDBMS, you may choose to use Windows authentication so that users do not have to remember their RDBMS login ID and password when logging in to the MicroStrategy system. With Windows authentication, users are automatically logged in to the MicroStrategy system using their Windows ID and password.

For detailed information about security views, see *Security views, page 91*.

 © 2016, MicroStrategy Inc.

To establish this configuration

1. In Developer, open the Project Source Manager, and on the Advanced tab, select **Use network login ID (Windows authentication)** as the Authentication mode.
2. From Web, log in as an administrator and select **Preferences**, select **Project Defaults**, select **Security**, and then enable **Windows Authentication** as the login mode.
3. In Developer, in the User Editor, expand **Authentication**, then select **Warehouse.**
4. Link users to their respective database user IDs using the **Warehouse passthrough Login** and **Warehouse passthrough password** boxes for each user. For details on each option, click **Help**.
5. Enable the setting for database execution to use linked warehouse logins on each project that you wish to use linked warehouse logins for database execution. To do this, right-click the project and select **Project Configuration**, expand the **Database instances** category, click **Execution**, and select the **Use linked warehouse login for execution** check box.

Connection maps: Standard authentication, connection maps, and partitioned fact tables

You may want to use this configuration if you implement access control policies in the RDBMS so that you can have multiple user accounts in the RDBMS, but not necessarily one for every user. In addition, you must use connection maps to enable report subscriptions if you are using Microsoft Analysis Services with integrated authentication.

For example, you are partitioning fact tables by rows, as described in *Splitting fact tables by rows, page 91*. You have a user ID for the 1st National Bank that only has access to the table containing records for that bank and another user ID for the Eastern Credit Bank that only has access to its corresponding table. Depending on the user ID used to log in to the RDBMS, a different table is used in SQL queries.

Although there are only a small number of user IDs in the RDBMS, there are many more users who access the MicroStrategy application. When users access the MicroStrategy system, they log in using their MicroStrategy user names and passwords. Using connection maps, Intelligence Server uses different database accounts to execute queries, depending on the user who submitted the report.

To establish this configuration

1. In Developer, open the Project Source Manager and click **Modify**.
2. On the Advanced tab, select **Use login ID and password entered by the user (standard authentication)** as the Authentication mode. This is the default setting.

© 2016, MicroStrategy Inc.

3. From Web, log in as an administrator and select **Preferences**, select **Project Defaults**, select **Security**, and then enable **Standard (user name & password)** as the login mode.
4. Create a database login for each of the RDBMS accounts.
5. Create a user group in the MicroStrategy system corresponding to each of the RDBMS accounts and then assign multiple users to these groups as necessary.
6. Define a connection mapping that maps each user group to the appropriate database login.

© 2016, MicroStrategy Inc.

4

Enabling Secure Communication

Introduction

SSL (secure socket layer) is an encryption technology that encodes communication over the Internet or local network so that only the recipient can read it.

This chapter describes the steps to enable secure, encrypted communications between MicroStrategy components using SSL.

The documentation below refers to third-party software, and thus is subject to change. MicroStrategy makes no guarantee on the availability or accuracy of third party documentation.

Overview of SSL and encryption in MicroStrategy

To enable private communications between MicroStrategy components, you can configure them to use SSL encryption while communicating with each other. You can use SSL to encrypt communications between the following applications:

- Intelligence Server and Developer, MicroStrategy Web, and Mobile Server
- MicroStrategy Web and users' browsers
- Mobile Server and MicroStrategy Mobile for iPhone, iPad, and Android
- Web Services and Office

In addition, you can set up encrypted communications between Intelligence Server and Web or Mobile Server, using AES encryption. For steps to set up encryption between

© 2016, MicroStrategy Inc.

Web or Mobile Server and Intelligence Server, see *Enabling encryption between Web or Mobile Server and Intelligence Server , page 196*.

The following is an overview of the tasks you need to perform to set up SSL for MicroStrategy applications:

- Obtain an SSL Certificate for the MicroStrategy applications. For steps, see *Obtaining SSL certificates, page 184*.
- Install the SSL certificates on the servers, where applicable. For steps, see *Installing the SSL certificates, page 187*.
- Configure MicroStrategy applications to require SSL access. For steps, see *Configuring the MicroStrategy applications to use SSL, page 190*.
- Test SSL access. For steps, see *Testing SSL access, page 194*.
- Configure client applications, such as Office and Mobile, to use the SSL HTTPS URL. For steps, see *Configuring MicroStrategy client applications to use an HTTPS URL, page 195*.

Obtaining SSL certificates

To enable secure communications between MicroStrategy components, you must obtain an SSL certificate for the following MicroStrategy applications, as applicable:

- Intelligence Server
- Developer (optional)
- MicroStrategy Web
- Mobile Server
- If you use Office, MicroStrategy Web Services

Use the following table to determine which applications you need to obtain SSL certificates for:

Requirement	Obtain SSL certificates for:
Secure communications between Intelligence Server and Developer	• Intelligence Server • Developer (optional)
Secure communications between MicroStrategy Web and users' browsers	• MicroStrategy Web
Secure communications between Mobile Server and Mobile for	• Mobile Server

© 2016, MicroStrategy Inc.

Requirement	Obtain SSL certificates for:
iPhone, iPad, and Android	
Secure communications between Web Services and MicroStrategy Office	• Web Services

The high-level overview of the process is as follows:

- Generate SSL Certificate Requests for the servers that the MicroStrategy components run on. For steps, see *Generating an SSL Certificate Signing Request, page 185*.
- Submit the Certificate Requests, and receive the corresponding certificates. For steps, see *Submitting a Certificate Request, page 186*.

Prerequisites

Before enabling SSL for your installed products, you require access to a Certificate Authority (CA), which may be one of the following:

- An enterprise CA, which is set up on your local network. The decision to set up an enterprise CA should be made at the executive level. To set up an enterprise CA on your network, refer to your server operating system's documentation.
- A commercial CA, such as VeriSign.
- If you need certificates for a development or proof-of-concept environment, you can set up your own CA to create self-signed certificates. For steps, see *Self-signed certificates: Creating a Certificate Authority for development, page 196*.

If you are using an enterprise CA or self-signed certificates for use with Android devices, it is recommended that you use devices that are running Android 4.0 (Ice Cream Sandwich) or better.

Generating an SSL Certificate Signing Request

You can use the OpenSSL utility to create an SSL Certificate Signing Request (CSR) for each of your applications.

If you are using a UNIX or Linux machine, the OpenSSL utility should be installed by default. If you are using a Windows machine, you can download the OpenSSL utility from http://www.openssl.org/.

To generate an SSL Certificate Signing Request using OpenSSL

1 Depending on your platform, do one of the following:

- **UNIX/Linux**: Open a terminal window.
- **Windows**: Open a command prompt window, and navigate to the location where OpenSSL is installed. By default, this is `C:\OpenSSL-Win32\bin`.

To generate a private key for the server

2 Type the following command, and press Enter:

```
openssl genrsa -des3 -out Server_key.key
```

Where *Server_key.key* is the name of the private key file. By default, the private key file is created in the current location. To create the file at a different location, replace *Server_key.key* with a path to create the new file.

You are prompted for a pass-phrase for the key.

3 Type a secure pass-phrase for the key, and press Enter. The key file is created.

To generate the Certificate Signing Request

4 Type the following command, and press Enter:

```
openssl req -new -key Server_key.key -out Server_
CSR.csr
```

Where *Server_key.key* is the private key file that you created, and *Server_CSR* is the CSR file.

5 You are prompted for information such as your organization's name, department name, country code, and so on. Type the information about your organization as you are prompted. When prompted for a Common Name, type the fully qualified domain name of the server that the application runs on. For example, if Intelligence Server runs on a machine called `intelligenceserver`, and your domain is `yourcompany.com`, the fully qualified domain name is `intelligenceserver.yourcompany.com`.

When you have entered all the required information, the CSR file is created.

6 Repeat this procedure for every application that you need a certificate for.

Submitting a Certificate Request

Once you have created a Certificate Request, you must submit it to a Certificate Authority (CA) to obtain an SSL certificate. If you are using a commercial CA, such as VeriSign, refer to their documentation for the procedure to submit the certificate.

If you have an enterprise CA running Microsoft Certificate Services on your local network, use the following procedure to submit a Certificate Request.

 © 2016, MicroStrategy Inc.

To submit a Certificate Request to an enterprise CA running Microsoft Certificate Services

1. Open the request file generated in *Generating an SSL Certificate Signing Request, page 185* with Microsoft Notepad, and copy the contents.
2. In your browser, open the following URL: `http://hostname/ CertSrv`, where *hostname* is the server on which Microsoft Certificate Services was installed. The Microsoft Certificate Services page opens.
3. Click **Request a Certificate**. The Choose Request Type page opens.
4. Click **Advanced Request** and click **Next**. The Submit a Certificate Request or Renewal Request page opens.
5. In the **Base64 Encoded Certificate Request** field, paste the contents of the request file, and click **Submit**. The Certificate Pending page is displayed.

   You may need to contact the administrator of the CA to ensure that the Certificate Request is approved.

6. Once the Certificate Request is approved, open the Certificate Services URL above.

   You must access the Certificate Services URL using the same machine and browser that you used to submit the certificate request.

7. Click **View the status of a pending certificate request**. The Pending Requests page opens.
8. Click the link for the certificate request you made. If the certificate has been issued, the Certificate Issued page opens.
9. Select the **Base 64 encoded** option, and click **Download certificate**. The certificate is downloaded.
10. Once you have downloaded the certificates for the applications you need, copy them to the machines where the applications are installed.

Installing the SSL certificates

You need to install the SSL certificates on the application servers that host the following applications:

- MicroStrategy Web, to enable secure communication between Web and users' browsers.

- MicroStrategy Mobile Server, to enable secure communication between Mobile Server and Mobile for iPhone, iPad, and Android.
- MicroStrategy Web Services, to enable secure communication between Web Services and Office.

When you have downloaded the SSL certificate from your enterprise or commercial CA, perform the following steps to install it on the application servers.

The following procedure describes the steps to install an SSL certificate in IIS 6 running on Microsoft Windows Server. If you have a different version of IIS, or are using a different application server, such as Tomcat, refer to the product's documentation for instructions to install SSL certificates.

To install the SSL certificates for Web, Mobile Server, or Web Services in IIS 6

1. From the **Start** menu, select **Settings**, then **Control Panel**, then select **Administrative Tools**. Open the **Internet Services Manager** tool.
2. On the left, navigate to the website that contains the application's virtual directory. By default, this is **Default Web Site**. Right-click the website and click **Properties**. The Properties dialog box opens.
3. On the **Directory Security** tab, click **Server Certificate**. The Web Server Certificate Wizard opens. Click **Next** to close the Welcome page.
4. Click **Process the pending request** and install the certificate. Click **Next**. The Process a Pending Request page opens.
5. Enter the path and file name of the issued certificate and click **Next**. The Certificate Summary page opens.
6. Review the information in the Certificate Summary, click **Next**, and then click **Finish** to install the certificate on the application server.

In addition, if you obtained the SSL certificate from an enterprise CA, you must add the CA as a trusted certificate authority using the steps described in *Adding your enterprise CA as a trusted certificate authority, page 188* below.

Adding your enterprise CA as a trusted certificate authority

If you used an enterprise CA on your network, or if you created a CA for demos, as described in *Self-signed certificates: Creating a Certificate Authority for development, page 196*, the CA's root certificate must be installed as a trusted root certificate authority in the following locations:

 © 2016, MicroStrategy Inc.

- The servers where Web, Mobile Server, or Web Services are installed, as applicable. For steps, see *To add your enterprise CA as a trusted certificate authority for Web, Mobile Server, or Web Services, page 189*.
- Mobile devices, such as iPhone, iPad, or Android phones and tablets. For steps to add a trusted CA for iPhone and iPad, see *To add your enterprise CA as a trusted certificate authority for iOS devices, page 190*.

For Android devices, you must add the certificate as a trusted certificate while creating a configuration for your devices in Mobile Administrator. For steps to configure Android devices, see the *Administering MicroStrategy Mobile* chapter in the MicroStrategy Mobile Design and Administration Guide.

Prerequisite

- You must have access to a copy of your enterprise CA's root certificate to perform this procedure.

  If your enterprise CA uses Microsoft Certificate Services, open the following URL in a browser window: `http://hostname/CertSrv`, where *hostname* is the computer on which Certificate Services is installed, click **Download a CA certificate, certificate chain, or CRL**, and under **Encoding method**, select **Base 64**. Click **Download CA certificate** and save it to the computer.

  If your enterprise CA uses OpenSSL, contact your administrator for a copy of the certificate.

To add your enterprise CA as a trusted certificate authority for Web, Mobile Server, or Web Services

1. On the machine where Web, Mobile Server, or Web Services is installed, from the **Start** menu, select **Run**, type `mmc`, and press Enter. The Microsoft Management Console opens.
2. From the **File** menu, select **Console**, and then select **Add/Remove Snap-in**. The Add/Remove Snap-in dialog box opens.
3. Click **Add** to open the Add Standalone Snap-in dialog box, click **Certificates**, and then click **Add**. The Certificates Snap-in dialog box opens.
4. Select **Computer Account** and click **Next**. The Select Computer page opens.
5. Click **Local Computer** and then click **Finish**. The Certificates snap-in is displayed in the list of selected snap-ins.
6. Click **OK** to return to the Console Root dialog box.
7. On the left, expand the **Certificates** snap-in, then expand Trusted Root Certificate Authorities.
8. Click **Action**, then **All Tasks**, then **Import**. The Certificate Import Wizard opens.
9. Click **Browse**, and select the certificate you downloaded from your CA.

10 Click **Next**. The Certificate Store page opens.

11 Select **Place all certificates in the following store**.

12 Click **Browse**, and select the **Trusted Root Certification Authorities** folder.

13 Click **Next**, then click **Finish**. A message is displayed, indicating that the import was successful.

To add your enterprise CA as a trusted certificate authority for iOS devices

1 To distribute your enterprise CA's root certificate to users, do one of the following:

- Send the certificate as an email attachment to all iOS users.
- On a server on your network, create a basic web page that allows users to download the certificate, and email the URL of the web page to your users.

The following steps must be performed for every iOS device in your organization.

2 On an iPhone or iPad, open the email or URL that contains the link to the certificate.

3 Tap the link to download the certificate. The certificate is downloaded, and is opened in the Install Profile dialog in the device's Settings screen.

4 In the Install Profile dialog, tap **Install**. A warning may be displayed, indicating that the authenticity of the certificate cannot be verified.

5 Click **Install**. The certificate is installed, and is shown as a trusted certificate.

If the device is protected with a passcode, you must type the passcode to install the certificate.

Configuring the MicroStrategy applications to use SSL

Use the following table to determine which MicroStrategy applications you need to configure SSL access for.

 © 2016, MicroStrategy Inc.

Requirement	Refer to:
Secure communications between Intelligence Server and Developer, MicroStrategy Web, or Mobile Server	*Configuring SSL between Intelligence Server and Developer, MicroStrategy Web, or Mobile Server, page 191*
Secure communications between Web and users' browsers	*Configuring Web, Mobile Server, and Web Services to require SSL access, page 193*
Secure communications between Mobile Server and Mobile for iPhone, iPad, and Android	*Configuring Web, Mobile Server, and Web Services to require SSL access, page 193*
Secure communications between Web Services and MicroStrategy Office	*Configuring Web, Mobile Server, and Web Services to require SSL access, page 193*

Configuring SSL between Intelligence Server and Developer, MicroStrategy Web, or Mobile Server

To enable Intelligence Server and Developer, MicroStrategy Web, or Mobile Server to communicate using SSL, you must first use the Configuration Wizard to add the SSL certificate to use for Intelligence Server, and then enable SSL in the respective applications.

Configuring SSL for Intelligence Server

Prerequisites

- You must have the SSL certificate you created for Intelligence Server.
- You must have the private key file that you created while requesting a certificate for Intelligence Server, as described in *Generating an SSL Certificate Signing Request, page 185*.

To configure SSL for Intelligence Server

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Configuration Wizard**. The Configuration Wizard opens.
2. On the Welcome screen, select **Configure Intelligence Server**, and click **Next**. The Metadata Connection page opens.
3. If you have previously configured Intelligence Server, click **Next** until you reach the SSL Configuration page. If this is the first time you are configuring Intelligence Server, click **Help** for instructions to configure Intelligence Server.
4. In the SSL Configuration page, enable the **Configure SSL** check box.
5. Click the button next to the **Certificate** field and browse to the certificate you created for Intelligence Server.

© 2016, MicroStrategy Inc.

6 Click the button next to the **Key** field and browse to the private key file you created while requesting the certificate for Intelligence Server.

7 In the **Password** field, type the password that you used while creating the private key for the certificate.

8 In the **SSL Port** field, type the port number to use for SSL access. By default, the port is `39321`.

Configuring SSL for Developer

Prerequisites

- You must use the Configuration Wizard to set up SSL for Intelligence Server, as described in *To configure SSL for Intelligence Server, page 191*.
- For additional security, you can enable Developer to verify Intelligence Server's certificate with the Certificate Authority (CA) before transmitting any data. If you want to enable this option, you must obtain the following:
  - Your CA's SSL certificate. If you are using a commercial CA, refer to their documentation for instructions to download their certificate.

    If you are using an enterprise CA that has Microsoft Certificate Services installed, visit `http://hostname/CertSrv`, where *`hostname`* is the computer on which Certificate Services is installed, and click **Download a CA certificate, certificate chain, or CRL**. Under **Encoding method**, select **Base64**.

To configure SSL for Developer

1 In Developer, right-click the server-based project source that you use to connect to Intelligence Server, and select **Modify Project Source**.

2 On the **Connection** tab, select the **Use SSL** check box.

3 If you want Developer to verify Intelligence Server's certificate with the CA every time a connection is made, select the **Verify Server Certificate** check box.

You must perform the following tasks to verify the server's certificate:

- Download the CA's certificate to the computer running Developer.
- In the **Client SSL Certificate Authority Certificate** field, enter the path to the CA's certificate. For example, `C:\Certificates\desktop.cer`.

4 Click **OK** to save the changes.

Configuring SSL for Web and Mobile Server

Prerequisites

© 2016, MicroStrategy Inc.

- You must use the Configuration Wizard to set up SSL for Intelligence Server, as described in *To configure SSL for Intelligence Server, page 191*.

To configure SSL for Web and Mobile Server

1. Open MicroStrategy Web Administrator or MicroStrategy Mobile Administrator, as applicable:
   - From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Web Administrator**. The MicroStrategy Web Administrator page opens.
   - From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Mobile Administrator**. The MicroStrategy Mobile Server Administrator page opens.
2. On the left, click **Security**. The Security page opens.
3. Under **Traffic to the Intelligence Server**, select the **SSL** option.
4. If you want Web or Mobile Server to verify Intelligence Server's SSL certificate, select **Validate Intelligence Server certificate**.
5. Click **Save** to save your changes.

Configuring Web, Mobile Server, and Web Services to require SSL access

You can configure your application server to require that clients, such as users' web browsers, access the following applications with SSL, using the HTTPS protocol:

- MicroStrategy Web, to enable secure communication between Web and users' browsers.
- MicroStrategy Mobile Server, to enable secure communication between Mobile Server and Mobile for iPhone, iPad and Android.
- MicroStrategy Web Services, to enable secure communication between Web Services and Office.

The following procedure describes the steps to require SSL access using IIS 6 running on Microsoft Windows Server. If you have a different version of IIS, or are using a different application server, such as Tomcat, refer to the product's documentation for instructions to require SSL for applications.

© 2016, MicroStrategy Inc.

To configure Web, Mobile Server, or Web Services to require SSL access

1. From the Windows **Start** menu, point to **Settings**, then **Control Panel**, and then select **Administrative Tools**. Open the **Internet Information Services Manager** tool.
2. On the left, navigate to the application's virtual directory. By default, these are:
   - Web: MicroStrategy
   - Mobile Server: MicroStrategyMobile
   - Web Services: MicroStrategyWS

   Right-click the virtual directory and click **Properties**. The Properties dialog box opens.
3. On the **Directory Security** tab, in the Secure Communications panel, click **Edit**. The Secure Communications dialog box opens.
4. Click **Require secure channel (SSL)** to require the use of HTTPS.
5. Click **OK**, then click **OK** again to close the Properties dialog box.
6. Repeat this procedure for the applications that apply to your requirements.
7. Close the Internet Information Services Manager.

Testing SSL access

You can perform the following steps to test SSL access to Web and Web Services.

To test SSL access to Web and Web Services

1. In your browser, enter the URL to access Web and Web Services. By default, these are:
   - Web (ASP.net): `http://hostname/MicroStrategy/asp/`, where *hostname* is the name of the server that Web is running on.
   - Web (J2EE): `http://hostname/MicroStrategy/servlet/mstrWeb`, where *hostname* is the name of the server that Web is running on.
   - Web Services: `http://hostname/MicroStrategyWS/MSTRWS.asmx`, where *hostname* is the name of the server that Web Services is running on.

© 2016, MicroStrategy Inc.

An error page should be displayed, with a 403.4 error indicating that SSL is required to access the page.

2 In the above URLs, replace `http://` with `https://`. After a short delay, Web should open, or the Web Services method list should be displayed (as applicable), indicating that the SSL access is working.

Configuring MicroStrategy client applications to use an HTTPS URL

To require iPhones, iPads, and Android devices to use HTTPS to connect to Mobile Server, you must update your device configurations in Mobile Server.

To require MicroStrategy Office to use SSL to connect to Web Services, in the Options dialog box, you must add the `https://` prefix to the URL for Web Services, as described in *To configure MicroStrategy Office to use SSL, page 195*.

To configure MicroStrategy Mobile for iPhone, iPad and Android to use SSL

1 Open the Mobile Administrator page.

2 Click **Mobile Configuration**. The Mobile Configuration page opens.

3 For the configuration you want to edit, click **Modify**. The configuration opens.

4 Click the **Connectivity Settings** tab.

5 For the Mobile Server that has SSL enabled, from the **Request Type** drop-down list, select **HTTPS**.

6 Click **Save** to save the configuration.

7 Repeat this procedure for every configuration that includes the above Mobile Server.

To configure MicroStrategy Office to use SSL

1 From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Office Configuration**.

2 Under **General**, select **Server**.

3 In the **Web Services URL** field, replace the `http://` prefix with `https://`.

4 Click **OK** to save the changes and close the MicroStrategy Office Options dialog box.

© 2016, MicroStrategy Inc.

Enabling encryption between Web or Mobile Server and Intelligence Server

You can also encrypt communication between the application and Intelligence Server, using AES encryption. Use the following procedure to enable encryption.

There is additional overhead involved in encrypting and decrypting the data between Web or Mobile Server and Intelligence Server, so you may experience a drop in performance. It is recommended that you use SSL to create a secure connection between Web or Mobile Server and Intelligence Server.

To enable encryption between Web or Mobile Server and Intelligence Server

1. Open the Administrator page for Web or Mobile Server, as applicable.
2. In the column on the left, click **Security**. The Security settings page opens.
3. Under **Traffic to the Intelligence Server**, select the **AES Tunneling** option.
4. Click **Save** to save your changes.

Self-signed certificates: Creating a Certificate Authority for development

If you are creating demos or proofs-of-concept that require SSL, you can set up a server that can act as a Certificate Authority (CA) to sign the certificates for the MicroStrategy applications.

It is recommended that you use self-signed certificates only in demo or development environments. Self-signed certificates are not recommended in a production environment for the following reasons:

- If the CA server is compromised, an attacker can use it to sign certificates for malicious sites.
- By default, users' devices and browsers do not accept self-signed certificates, which may cause users to receive security warnings and disrupt their workflows.

© 2016, MicroStrategy Inc.

You can set up a CA server using the OpenSSL utility. If you are using a UNIX or Linux machine, OpenSSL should be installed by default. If you are using a Windows machine, you can download the OpenSSL utility from http://www.openssl.org/.

To set up a CA, you need to perform the following tasks:

- Create the directories and configuration files for the CA. See *Creating the directories and configuration files for your CA, page 197*.
- Create the server's private key and root certificate. See *Creating the private key and root certificate for the CA, page 198*.
- Add the root certificate as a trusted certificate on your network. See *Adding your enterprise CA as a trusted certificate authority, page 188*.
- Configure OpenSSL to use the server's private key and certificate to sign certificate requests. See *Configuring OpenSSL to use your private key and root certificate, page 199*.
- Create certificates for the MicroStrategy applications. See *Signing certificate requests using your CA, page 200*.

Creating the directories and configuration files for your CA

To create your CA using OpenSSL, you must create directories to store important files for the CA, such as the server's private keys, certificates that have been signed, and so on. In addition, you must create the files that track the certificates that have been created, and an OpenSSL configuration file for your CA.

To create the directories and files for the CA

1 Using Windows Explorer or the UNIX Terminal, as applicable, create the following directories:

Directory	Folder name
A root directory for the CA.	A name of your choice. For example, `devCA`
A subdirectory to store the CA's private key	`private` For example, `devCA/private`
A subdirectory to store new certificates issued by the CA	`certs` For example, `devCA/certs`
A subdirectory to store the new certificates in an unencrypted format	`newcerts` For example, `devCA/newcerts`

2 In the root directory for the CA, use a text editor to create the following files:

Filename	Description
`serial` (no extension)	Contains the serial number for the next certificate. When you create the file, you must add the serial number for the first certificate. For example, `01`.
`index.txt`	Used as a database to track certificates that have been issued.

3 Depending on your platform, do one of the following:

- **UNIX/Linux**: Open a terminal window, and navigate to the location where OpenSSL is installed.

  The default installation folder may depend on the distribution you are using. For example, for Red Hat Enterprise Linux, the default folder is `/etc/pki/tls`.

- **Windows**: Open a command prompt window, and navigate to the location where OpenSSL is installed. By default, this is `C:\OpenSSL-Win32\bin`.

4 Create a copy of the OpenSSL configuration file `openssl.cnf`, and paste it in the root directory you created for your CA. Use a different file name, for example, `openssl.dev.cnf`.

Creating the private key and root certificate for the CA

Once you have set up the files and directories for your CA, you can create a root certificate, which is used to sign certificate requests from MicroStrategy applications.

This procedure assumes that you have followed all the steps in *Creating the directories and configuration files for your CA, page 197*.

To create the private key and root certificate for the CA

1 Depending on your platform, do one of the following:

- **UNIX/Linux**: Open a terminal window.
- **Windows**: Open a command prompt window, and navigate to the location where OpenSSL is installed. By default, this is `C:\OpenSSL-Win32\bin`.

2 To create the private key and root certificate, type the following command, and press Enter:

```
openssl req -config devCApath/openssl.dev.cnf -new -
x509 -extensions v3_ca -keyout
```

 © 2016, MicroStrategy Inc.

```
devCApath/private/devCA.key -out
devCApath/certs/devCA.crt -days 1825
```

Substitute the values in italics based on the following:

- *devCApath*: The root directory for your CA, which is created as part of the procedure described in *Creating the directories and configuration files for your CA, page 197*. For example, `/etc/pki/tls/devCA`.
- *openssl.dev.cnf*: The copy of the default OpenSSL configuration file, created in the root directory for your CA.
- *devCA.key*: The filename for the private key.
- *devCA.crt*: The filename for the root certificate.

3 You are prompted for a pass-phrase for the key, and for information about your CA, such as your location, organization name, and so on. Use a strong pass-phrase to secure your private key, and type the required information for the CA. The private key and root certificate are created.

Configuring OpenSSL to use your private key and root certificate

To start creating certificates for the MicroStrategy applications in your development environment, you must configure OpenSSL to use your CA's private key and root certificate to sign certificate requests. For information on creating certificate requests for applications, see *Generating an SSL Certificate Signing Request, page 185*.

Prerequisites

This procedure assumes that you have completed the following steps:

- Create the files and directory structure for your CA, including a copy of the default OpenSSL configuration file, as described in *Creating the directories and configuration files for your CA, page 197*.
- Create a private key and root certificate for your CA, as described in *Creating the private key and root certificate for the CA, page 198*.

To configure OpenSSL to use your CA's root certificate

1 Use a text editor, such as Notepad, to open the copy of the OpenSSL configuration file in your CA's root directory. For example, `openssl.dev.cnf`.

2 Scroll to the `CA_default` section, and edit the following values:

- `dir`: Change this value to the root folder that you created for your CA. For example, `/etc/pki/tsl/devCA`.

- `certificate`: Change this value to `$dir/certs/devCA.crt`, where `devCA.crt` is the root certificate that you created for your CA.
- `private_key`: Change this value to `$dir/private/devCA.key`, where `devCA.key` is the private key that you created for your CA.

3 Save the file.

Signing certificate requests using your CA

Once you have configured OpenSSL to use your CA's private key and root certificate, you can sign certificate requests to create the SSL certificates for the MicroStrategy applications. The steps to create certificates follow.

Prerequisites

This procedure assumes that you have completed the following steps:

- Create the files and directory structure for your CA, including a copy of the default OpenSSL configuration file, as described in *Creating the directories and configuration files for your CA, page 197*.
- Create a private key and root certificate for your CA, as described in *Creating the private key and root certificate for the CA, page 198*.
- Configure OpenSSL to use the private key and root certificate, as described in *Configuring OpenSSL to use your private key and root certificate, page 199*.
- Create a certificate signing request (CSR file) for the applications that require SSL certificates, as described in *Generating an SSL Certificate Signing Request, page 185*. Copy the CSR file to the server that hosts your CA.

To sign certificate requests using your CA

1 Depending on your platform, do one of the following:

- **UNIX/Linux**: Open a terminal window, and navigate to the location where OpenSSL is installed.

  The default installation folder may depend on the distribution you are using. For example, for Red Hat Enterprise Linux, the default folder is `/etc/pki/tls`.

- **Windows**: Open a command prompt window, and navigate to the location where OpenSSL is installed. By default, this is `C:\OpenSSL-Win32\bin`.

2 Type the following command, and press Enter:

 © 2016, MicroStrategy Inc.

```
openssl ca -config devCApath/openssl.dev.cnf -policy
policy_anything -out devCApath/certs/mstrapp.crt -
infiles CSRpath/mstrapp.csr
```

Substitute the values in italics based on the following:

- *devCApath*: The root directory for your CA, which is created as part of the procedure described in *Creating the directories and configuration files for your CA, page 197*. For example, `/etc/pki/tls/devCA`.
- *openssl.dev.cnf*: The OpenSSL configuration file for your CA, configured to use your CA's private key and root certificate, as described in *Configuring OpenSSL to use your private key and root certificate, page 199*.
- *mstrapp.crt*: The filename for the certificate to be generated for the MicroStrategy application.
- *CSRpath*: The folder where the certificate signing request is stored.
- *mstrapp.csr*: The certificate signing request for the MicroStrategy application.

The certificate is generated, and is stored in the `certs` folder.

3. Copy the generated certificate to the machine where the MicroStrategy application is hosted.
4. Repeat this procedure for all MicroStrategy applications that require SSL certificates.

 © 2016, MicroStrategy Inc.

5

MANAGING YOUR LICENSES

Introduction

As a system administrator, it is important that you manage your MicroStrategy product licenses to maintain license compliance. Managing your licenses can also help you take full advantage of your licenses. For example, you might have a CPU-based Intelligence Server license for four CPUs, but only be using two CPUs. An audit of your licenses can alert you to this issue and you can then modify your setup so that you use all four of your licensed CPUs.

This chapter covers how to manage the licenses involved in your MicroStrategy system. Topics include:

- *Managing and verifying your licenses, page 203*
- *Auditing and updating licenses, page 206*
- *Updating CPU affinity, page 210*

Managing and verifying your licenses

MicroStrategy licenses are managed differently according to the license type that is purchased. Refer to your MicroStrategy contract and any accompanying contract documentation for descriptions of the different MicroStrategy license types.

MicroStrategy uses two main categories of licenses:

© 2016, MicroStrategy Inc.

- *Named User licenses, page 204*, in which the number of users with access to specific functionality are restricted
- *CPU licenses, page 205*, in which the number and speed of the CPUs used by MicroStrategy server products are restricted

MicroStrategy License Manager can assist you in administering your MicroStrategy licenses. For information about License Manager, see *Using License Manager, page 207*.

When you obtain additional licenses from MicroStrategy, use License Manager to update your license information. For details, see *Updating your license, page 209*.

Named User licenses

In a Named User licensing scheme, the privileges given to users and groups determine what licenses are assigned to users and groups. Intelligence Server monitors the number of users in your MicroStrategy system with each privilege, and compares that to the number of available licenses.

For example, the Web Use Filter Editor privilege is a Web Professional privilege. If you assign this privilege to User1, then Intelligence Server grants a Web Professional license to User1. If you only have one Web Professional license in your system and you assign any Web Professional privilege, for example Web Edit Drilling And Links, to User2, Intelligence Server displays an error message when any user attempts to log in to MicroStrategy Web.

The Administrator user that is created with the repository is not considered in the licensed user count.

To fix this problem, you can either change the user privileges to match the number of licenses you have, or you can obtain additional licenses from MicroStrategy. License Manager can determine which users are causing the metadata to exceed your licenses and which privileges for those users are causing each user to be classified as a particular license type (see *Using License Manager, page 207*).

For more information about the privileges associated with each license type, see the *List of Privileges* chapter in the Supplemental Reference for System Administration. Each privilege group has an introduction indicating any license that the privileges in that group are associated with. Users without any product-based privileges are listed in License Manager in the group **Users without license association**, and are not counted against any MicroStrategy licenses.

Note the following:

- Privileges that relate to OLAP Services, Report Services, and Distribution Services licenses are present in multiple privilege groups. These privileges are marked with asterisks, and are listed at the top of each group's list of privileges.
- Only users who have the Use Developer privilege in the Analyst group are granted Analyst or Developer licenses. Users who do not have the Use Developer

 © 2016, MicroStrategy Inc.

privilege are not granted either of these licenses, even if they have all other privileges from these privilege groups.

Verifying Named User licenses

To verify your Named User licenses, Intelligence Server scans the metadata repository daily for the number of users fitting each Named User license type. If the number of licenses for a given type has been exceeded, an error message is displayed when a user logs in to a MicroStrategy product. Contact your MicroStrategy account executive to increase your number of Named User licenses. For detailed information on the effects of being out of compliance with your licenses, see *Effects of being out of compliance with your licenses, page 206*.

For steps to manually verify your Named User licenses using License Manager, see *Auditing your system for the proper licenses, page 208*.

You can configure the time of day that Intelligence Server verifies your Named User licenses.

To configure the time when Named User licenses are verified

1. In Developer, right-click a project source and select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
2. Expand the **Server** category, and select **Advanced**.
3. Specify the time in the **Time to run license check (24 hr format)** field.
4. Click **OK** to accept any changes and close the Intelligence Server Configuration Editor.

CPU licenses

When you purchase licenses in the CPU format, the system monitors the number of CPUs being used by Intelligence Server in your implementation and compares it to the number of licenses that you have. You cannot assign privileges related to certain licenses if the system detects that more CPUs are being used than are licensed. For example, this could happen if you have MicroStrategy Web installed on two dual-processor machines (four CPUs) and you have a license for only two CPUs.

To fix this problem, you can either use License Manager to reduce the number of CPUs being used on a given machine so it matches the number of licenses you have, or you can obtain additional licenses from MicroStrategy. To use License Manager to determine the number of CPUs licensed and, if necessary, to change the number of CPUs being used, see *Using License Manager, page 207*.

The ability to deploy Intelligence Server or MicroStrategy Web on specific, selected CPUs (a subset of the total number of physical CPUs) on a given machine is called CPU affinity. For details on setting up CPU affinity, see *Updating CPU affinity, page 210*.

© 2016, MicroStrategy Inc.

Verifying CPU licenses

To verify your CPU licenses, Intelligence Server scans the network to count the number of CPUs in use by Intelligence Servers. If the number of CPU licenses has been exceeded, an error message is displayed when a user logs in to a MicroStrategy product. Contact your MicroStrategy account executive to increase your number of CPU licenses. For detailed information on the effects of being out of compliance with your licenses, see *Effects of being out of compliance with your licenses, page 206*.

For steps to manually verify your CPU licenses using License Manager, see *Auditing your system for the proper licenses, page 208*.

Effects of being out of compliance with your licenses

If your system is determined to be out of compliance with your licenses, an error message is displayed any time a user accesses an administrative product, such as the MicroStrategy Web Administrator page or the Administration icon in Developer. This message describes the specific types of licenses that are not in compliance and states how many days remain before Intelligence Server can no longer be restarted. This error message is only a warning, and users can still use the administrative product.

After the system has been out of compliance for fifteen days, an additional error message is displayed to all users when they log into a project source, warning them that the system is out of compliance with the available licenses. This error message is only a warning, and users can still log in to the project source.

After the system has been out of compliance for thirty days, Intelligence Server can no longer be restarted once it is shut down. In addition, if the system is out of compliance with Named User licenses, the privileges associated with the out-of-compliance products are disabled in the User Editor, Group Editor, and Security Role Editor to prevent them from being assigned to any additional users.

Contact your MicroStrategy account executive to purchase additional licenses. For information on how Intelligence Server verifies licenses, see *Verifying Named User licenses, page 205* and *Verifying CPU licenses, page 206*.

Auditing and updating licenses

Once your MicroStrategy system is in place, Intelligence Server verifies how your system is being used in relation to licenses and users. You can use License Manager to ensure that your system is in compliance with your licenses.

You can check for and manage the following licensing issues:

- More copies of a MicroStrategy product are installed and being used than you have licenses for.
- More users are using the system than you have licenses for.

 © 2016, MicroStrategy Inc.

- More CPUs are being used with Intelligence Server than you have licenses for.

Using License Manager

License Manager is a tool for auditing and administering your MicroStrategy licenses and installation. You can run License Manager as a graphical user interface (GUI) or as a command line tool, in either Windows or UNIX/Linux environments.

In both GUI mode and command line mode, License Manager allows you to:

- Audit your MicroStrategy products.
- Request an Activation Code and activate your MicroStrategy installation.
- Update your license key.

Additionally, in GUI mode License Manager allows you to:

- Determine the number of product licenses in use by a specified user group.
- Display the enabled or disabled licenses used by a particular user group for selected products.

  From this information, you can determine whether you have the number of licenses that you need. You can also print a report, or create and view a Web page with this information.
- Update licenses by providing the new license key, without re-installing the products. For example, you can:
  - Upgrade from an evaluation edition to a standard edition.
  - Update the number of Intelligence Server processors allowed.
  - Update the processor speed allowed.
- Activate or deactivate your MicroStrategy installation.

  For more information on activating your MicroStrategy installation, see the Installation and Configuration Guide.
- Change the number of CPUs being used for a given MicroStrategy product, such as Intelligence Server or MicroStrategy Web, if your licenses are based on CPUs.
- Trigger a license verification check after you have made any license management changes, so the system can immediately return to normal behavior.
- View your machine's configuration including hardware and operating system information.
- View your MicroStrategy installation history including all license keys that have been applied.

© 2016, MicroStrategy Inc.

- View the version, edition, and expiration date of the MicroStrategy products installed on the machine.

If the edition is not an Evaluation edition, the expiration date has a value of "Never."

For detailed steps to perform all of these procedures, see the License Manager Help (from within License Manager, press **F1**).

To start License Manager

License Manager can be run on Windows or UNIX, in either GUI mode or command line mode.

- **Windows GUI**: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **License Manager**. License Manager opens in GUI mode.
- **Windows command line**: From the **Start** menu, select **Run**. Type `CMD` and press ENTER. A command prompt window opens. Type `malicmgr` and press ENTER. License Manager opens in command line mode, and instructions on how to use the command line mode are displayed.
- **UNIX/Linux GUI**: In a UNIX or Linux console window, browse to `<HOME_PATH>` where `<HOME_PATH>` is the directory that you specified as the home directory during installation. Browse to the folder `bin` and type `./mstrlicmgr`, then press ENTER. License Manager opens in GUI mode.
- **UNIX/Linux command line**: In a UNIX or Linux console window, browse to `<HOME_PATH>` where `<HOME_PATH>` is the directory that you specified as the home directory during installation. Browse to the folder `bin` and type `./mstrlicmgr -console`, then press ENTER. License Manager opens in command line mode, and instructions on how to use the command line mode are displayed.

Auditing your system for the proper licenses

License Manager counts the number of licenses based on the number of users with at least one privilege for a given product. The Administrator user that is created by default with the repository is not considered in the count.

To audit your system, perform the procedure below on each server machine in your system.

Note the following:

- In rare cases, an audit can fail if your metadata is too large for the Java Virtual Machine heap size. For steps to modify the Java Virtual Machine heap size in

 © 2016, MicroStrategy Inc.

your system registry settings, see MicroStrategy Tech Notes TN6446 and TN30885.

- If you are using License Manager on the physical machine on which Intelligence Server is installed, and a three-tier project source does not exist on that machine, you cannot log in to the server. To audit your licenses in this case, you must first create a three-tier project source pointing to the Intelligence Server. You can use either MicroStrategy Configuration Wizard or Developer's Project Source Manager to create this project source.

To audit your MicroStrategy licenses

1. Open MicroStrategy License Manager. For instructions, see *To start License Manager, page 208*.

   In command line mode, the steps to audit licenses vary from those below. Refer to the License Manager command line prompts to guide you through the steps to audit licenses.

2. On the **Audit** tab, expand the Intelligence Server folder.
3. Double-click a project source name (PSN). A Login dialog box opens.
4. Type your MicroStrategy login and password for the selected Intelligence Server and click **Connect**. If you are in compliance, a message appears notifying you that you are in compliance with your software license agreement. Click **OK** to proceed.
5. Select the **Everyone** group and click **Audit**. A folder tree of the assigned licenses is listed in the **Number of licenses** pane.

   Users with no product-based privileges are listed under **Users without license association**.

6. Count the number of licenses per product for enabled users. Disabled users do not count against the licensed user total, and should not be counted in your audit.
7. To print the summary information, click **Print**.

   For detailed information, click **Report** to create and view XML, HTML, and CSV reports. You can also have the report display all privileges for each user based on the license type. To do this, select the **Show User Privileges in Report** check box.

8. Total the number of users with each license across all machines.

Updating your license

If you need to update a license and you receive a new license key from MicroStrategy, use the License Manager to perform the upgrade. If you have licenses based on the number of CPUs being used, you can also use the update process to change the number of CPUs

being used by a given product. For complete details on performing an upgrade in your environment, see the Upgrade Guide.

You must update your license key on all machines where MicroStrategy products are installed. License Manager updates the license information for the products that are installed on that machine.

To update a MicroStrategy license

1. Acquire a new license key from MicroStrategy.
2. Open MicroStrategy License Manager. For instructions, see *To start License Manager, page 208*.

   In command line mode, the steps to update your license vary from those below. Refer to the License Manager command line prompts to guide you through the steps to update your license.

3. On the **License Administration** tab, select the **Update local license key** option and click **Next**.
4. Type or paste the new key in the **New License Key** field and click **Next**.

   If you have one or more products that are licensed based on CPU usage, the **Upgrade** window opens, showing the maximum number of CPUs each product is licensed to use on that machine. You can change these numbers to fit your license agreement. For example, if you purchase a license that allows more CPUs to be used, you can increase the number of CPUs being used by a product.

5. The results of the upgrade are shown in the Upgrade Results dialog box. License Manager can automatically request an Activation Code for your license after you update.
6. If you have updated your license information, restart Intelligence Server after the update. This allows the system to recognize the license key update and system behavior can return to normal.

Updating CPU affinity

Depending on the number of CPU-based licenses you purchase, you can have multiple processors (CPUs) running Intelligence Server and MicroStrategy Web. The ability to deploy Intelligence Server or MicroStrategy Web on specific, selected CPUs (a subset of the total number of physical CPUs) on a given machine is called CPU affinity (or processor affinity). As part of the installation process you must provide the number of processors to be used by Intelligence Server or MicroStrategy Web on that machine.

 © 2016, MicroStrategy Inc.

CPU affinity for Intelligence Server on Windows

Upon installation, if the target machine contains more than one physical processor and the MicroStrategy license key allows more than one CPU to run Intelligence Server, you are prompted to provide the number of CPUs to be deployed. The upper limit is either the number of licensed CPUs or the physical CPU count, whichever is lower.

After installation you can specify CPU affinity through the MicroStrategy Service Manager. This requires administrator privileges on the target machine.

To change CPU affinity settings in Service Manager

1. On the machine whose CPU affinity you want to change, from the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Service Manager**. Service Manager opens.
2. From the **Service** drop-down list, select **MicroStrategy Intelligence Server**.
3. Click **Options**. The Service Options dialog box opens.
4. Select the **Intelligence Server Options** tab.
5. In the **Processor Usage** section, select which processors Intelligence Server should use.
6. Click **OK**. The Service Options dialog box closes and CPU affinity has been changed.

CPU affinity for Intelligence Server on UNIX/Linux

CPU affinity behaves in a similar manner in both Windows and UNIX/Linux environments. This section describes details for setting up CPU affinity for running Intelligence Server Universal Edition in a UNIX or Linux environment. Information specific to AIX and Solaris is also discussed.

The ability to set CPU affinity on UNIX/Linux requires special system-level privileges. MicroStrategy must be run under the root UNIX/Linux account, otherwise an error message appears.

If the target machine contains more than one physical processor and the MicroStrategy license key allows more than one CPU to run Intelligence Server Universal Edition, you are prompted to provide the number of CPUs to be deployed. The upper limit is either the number of licensed CPUs or the physical CPU count, whichever is lower.

Each UNIX/Linux platform exposes its own set of functionality to bind processes to processors. However, UNIX/Linux also provides commands to easily change the processor assignments. As a result, Intelligence Server periodically checks its own CPU affinity and takes steps whenever the CPU affinity mask does not match the overall CPU licensing. Whenever your licenses do not match your deployment, CPU affinity is automatically adjusted to the number of CPUs necessary to be accurate again.

© 2016, MicroStrategy Inc.

This automatic adjustment for CPU affinity attempts to apply the user's specified CPU affinity value when it adjusts the system, but it may not always be able to do so depending on the availability of processors. For example, if you own two CPU licenses and CPU affinity is manually set to use Processor 1 and Processor 2, the CPU affinity adjustment may reset CPU usage to Processor 0 and Processor 1 when the system is automatically adjusted.

Under HP-UX, Processor 0 is always grayed out and cannot be assigned to Intelligence Server. This is because in HP-UX, Processor 0 is always assigned to the default processor set, and cannot be reassigned to any other user-created processor set. Intelligence Server uses processor sets to implement multi-CPU Processor Affinity. The only times Intelligence Server can use Processor 0 are when it is set to "all CPUs," or when it is set to "one CPU." For additional detail about processor sets in HP-UX, see *Processor sets, page 213*.

Changing CPU affinity in UNIX/Linux

You can specify CPU affinity either through the MicroStrategy Service Manager, or by modifying Intelligence Server options. If you want to view and modify Intelligence Server's options, it must be registered as a service. You can register Intelligence Server Universal as a service using the Configuration Wizard by selecting the Register Intelligence Server as a Service option; alternatively, you can follow the procedure below.

To set up Intelligence Server to run as a service

1. Navigate to the `bin` directory in the installation location.
2. Type the following command:

   `mstrctl -s IntelligenceServerName rs`

Whenever you change the CPU affinity, you must restart the machine.

CPU affinity implementation on AIX

This section describes resource sets, Intelligence Server startup and shutdown, and using a non-root account.

Resource sets

On an AIX system, CPU affinity is implemented using resource sets. A resource set is a logical structure that contains a list of specific CPUs that will be used. Processors are bound to a resource set. Resource sets do not define exclusive use of a resource; the same CPU can be part of several different resource sets.

 © 2016, MicroStrategy Inc.

Intelligence Server startup and shutdown

When Intelligence Server starts, it creates a resource set, assigns CPUs to that resource set, and binds the server process to that resource set. Resource sets exist only for the lifetime of the process. If the process shuts down, whether normally or unexpectedly, the resource set is cleaned from the system.

Using a non-root account

While the root account is mandatory for installing MicroStrategy with a CPU-based license, it can be possible to run Intelligence Server as a non-root account. If this occurs, you must manually define the resource set to be used by Intelligence Server because the automatic adjustments to correct CPU affinity cannot be made.

For example, consider a scenario in which Intelligence Server only has a license for three CPUs, and is running from a non-root account. If the user assigns a resource set to Intelligence Server that has five CPUs, Intelligence Server attempts to use all five CPUs because it cannot restrict itself to the required three CPUs when running under a non-root account. This results in an out-of-compliance situation.

To fix this problem, either run Intelligence Server from a root account, or do not assign a resource set to Intelligence Server that contains more CPUs than your CPU license allows.

CPU affinity implementation on Solaris and HP-UX

This section describes processor sets, Intelligence Server startup and shutdown, and how to specify a processor set.

Processor sets

On a Solaris or HP-UX system, multi-CPU affinity is implemented using processor set binding. A processor set is a collection of processors. A process assigned to a processor set can only use the CPUs specified for that processor set. Additionally, a processor set takes exclusive ownership of the CPUs assigned to it. Only the processes assigned to the processor set can use the processor set's CPUs. Other processes are not allowed to use any of these CPUs.

A processor set exists beyond the lifetime of the process that created it. Therefore, when a process is shut down, the process must delete the processor set that was created. For example, if a process creates a processor set with three CPUs and the process unexpectedly terminates without deleting the processor set it created, the three CPUs cannot be utilized by any other process until the system is rebooted or the processor set is manually deleted.

Whenever the CPU license is limited to a single CPU, Intelligence Server dynamically binds the process to the selected CPU without creating a processor set.

Intelligence Server startup and shutdown

When Intelligence Server starts, it creates a processor set, assigns CPUs to that processor set, and binds the server process to that processor set. However, Intelligence Server creates processor sets only on machines that have more than two physical CPUs and only when the number of physical CPUs is greater than the number of CPUs licensed. For example, if Intelligence Server licensed for two CPUs is installed on a four-CPU machine, a processor set containing two CPUs is created. However, if Intelligence Server licensed for four CPUs is installed on the same machine, the processor set is not created, because the number of physical CPUs is not greater than the number of licensed CPUs.

Intelligence Server deletes the processor set before shutting down so that the related processes do not remain locked. If Intelligence Server terminates unexpectedly, when restarted it performs a cleanup of the processor set it had created. However, if Intelligence Server is not restarted immediately after termination, you may need to manually delete the processor set so the CPUs are free to be used by other applications.

Specifying a processor set

You can specify an existing processor set for Intelligence Server to use. To do so, type the following command:

```
Prset -e <processorsetID> mstrsvr
```

Intelligence Server may automatically modify the existing processor set to keep your licenses mapped properly, if necessary.

For example, consider a scenario where an existing processor set is comprised of Processor 0, Processor 1, and Processor 2. A CPU-based license for Intelligence Server allows two physical CPUs to be used. Intelligence Server is installed and configured to use this existing processor set at startup. In this scenario, Intelligence Server modifies the existing processor set to use only two physical CPUs so that it matches its license. Intelligence Server does not create a new processor set, and when it shuts down it does not delete this processor set.

CPU affinity for MicroStrategy Web

If you have CPU-based licenses for MicroStrategy Web, the CPU affinity feature allows you to match your CPUs and licenses by choosing which processors MicroStrategy Web uses on a given machine.

This feature is only available in the ASP.NET version of MicroStrategy Web. It is not implemented in MicroStrategy Web Universal.

This section describes settings that may interact with CPU affinity that you must consider, and provides steps to update CPU affinity in your environment.

 © 2016, MicroStrategy Inc.

CPU affinity and IIS

Before configuring CPU affinity for MicroStrategy Web, you should understand how the CPU affinity setting behaves on different configurations of IIS, and how it interacts with other IIS settings such as the Web Garden mode.

IIS versions

CPU affinity can be configured on machines running IIS 6.0 or 7.0. The overall behavior depends on how IIS is configured. The following cases are considered:

- **Worker process isolation mode**: In this mode, the CPU affinity setting is applied at the application pool level. When MicroStrategy Web CPU affinity is enabled, it is applied to all ASP.NET applications running in the same application pool. By default, MicroStrategy Web runs in its own application pool. The CPU affinity setting is shared by all instances of MicroStrategy Web on a given machine. Worker process isolation mode is the default mode of operation on IIS 6.0 when the machine has not been upgraded from an older version of Windows.
- **IIS 5.0 compatibility mode**: In this mode, all ASP.NET applications run in the same process. This means that when MicroStrategy Web CPU affinity is enabled, it is applied to all ASP.NET applications running on the Web server machine. A warning is displayed before installation or before the CPU affinity tool (described below) attempts to set the CPU affinity on a machine with IIS running in IIS 5.0 compatibility mode.

This is the default mode of operation when the machine has been upgraded from an older version of Windows.

Web Garden mode

Both IIS 6.0 and IIS 7.0 support a "Web Garden" mode, in which IIS creates some number of processes, each with affinity to a single CPU, instead of creating a single process that uses all available CPUs. The administrator specifies the total number of CPUs that are used. The Web Garden settings can interact with and affect MicroStrategy CPU affinity.

The Web Garden setting should not be used with MicroStrategy Web. At runtime, the MicroStrategy Web CPU affinity setting is applied after IIS sets the CPU affinity for the Web Garden feature. Using these settings together can produce unintended results.

In both IIS 6.0 and IIS 7.0, the Web Garden feature is disabled by default.

CPU affinity interaction depends on how IIS is configured, as described below:

- In worker process isolation mode, the Web Garden setting is applied at the application pool level. You specify the number of CPUs to be used. A given number of CPUs are specified, and IIS creates that number of `w3wp.exe` instances. Each of the instances runs all of the ASP.NET applications associated with the application pool.

© 2016, MicroStrategy Inc.

The Web Garden feature is configured through the application pool settings. For more information, refer to your IIS documentation.

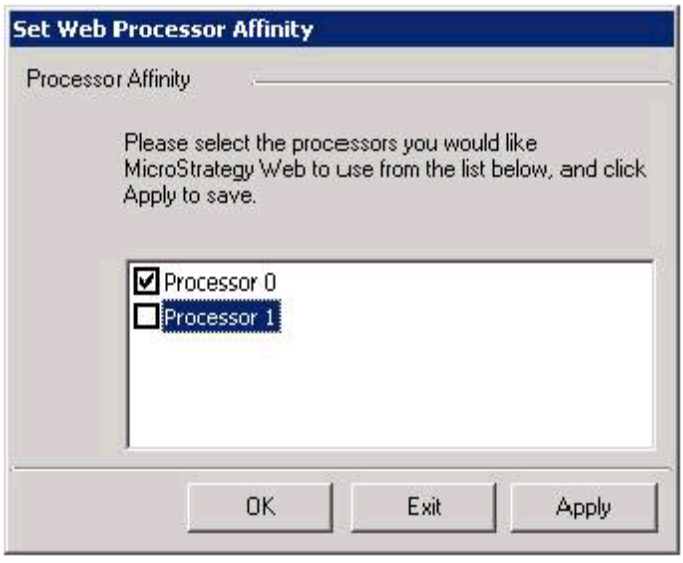
- In IIS 5.0 compatibility mode, a single setting affects all ASP.NET applications. The Web Garden feature is enabled or disabled using the **WebGarden** and **cpuMask** attributes under the processModel node in machine.config. A given number of CPUs are specified in the mask, and IIS creates that number of `aspnet_wp.exe` instances. Each of these instances runs the ASP.NET applications. For more information, refer to your IIS documentation.

IIS provides metabase properties (SMPAffinitized and SMPProcessorAffinityMask) to determine the CPU affinity for a given application pool. Do not use these settings in conjunction with the MicroStrategy Web CPU affinity setting.

Updating CPU affinity changes

After MicroStrategy Web is installed in your environment, you can update MicroStrategy Web's CPU affinity using a tool called `MAWebAff.exe`. This tool is located in the root directory of the MicroStrategy Web application, which is located by default at `C:\Program Files (x86)\MicroStrategy\Web ASPx`. The `MAWebAff.exe` tool allows you to choose the physical CPUs MicroStrategy Web can use. The number of CPUs that can be used depends on the limit specified by the license.

The `MAWebAff.exe` tool is shown below:



The `MAWebAff.exe` tool lists each physical CPU on a machine. You can add or remove CPUs or disable CPU affinity using the associated check boxes. Clearing all check boxes prevents the MicroStrategy Web CPU affinity setting from overriding any IIS-related CPU affinity settings.

 © 2016, MicroStrategy Inc.

To update CPU affinity

1. Double-click the `MAWebAff.exe` tool to open the CPU affinity tool.
2. Select or clear the check boxes for each processor as desired.
3. Click **Apply** to apply the settings without closing the tool, or click **OK** to apply settings and close the tool.

Clicking **Exit** closes the tool without saving any settings.

4. Restart IIS to apply your CPU affinity changes.

 © 2016, MicroStrategy Inc.

6

MANAGING YOUR PROJECTS

Introduction

In a MicroStrategy system, a project is the environment in which reporting is done. A project:

- Determines the set of data warehouse tables to be used, and therefore the set of data available to be analyzed.
- Contains all schema objects used to interpret the data in those tables. Schema objects include objects such as facts, attributes, and hierarchies.
- Contains all application objects used to create reports and analyze the data. Application objects include objects such as reports, metrics, and filters.
- Defines the security scheme for the user community that accesses these objects. Security objects include objects such as security roles, privileges, and access control lists.

The recommended methodology and tools for managing projects in the MicroStrategy system include:

- *The project life cycle, page 220*
- *Implementing the recommended life cycle, page 223*
- *Duplicating a project, page 225*
- *Updating projects with new objects, page 230*
- *Copying objects between projects: Object Manager, page 232*
- *Merging projects to synchronize objects, page 266*
- *Comparing and tracking projects, page 272*

© 2016, MicroStrategy Inc.

- *Deleting unused schema objects: managed objects, page 276*

For information about creating a project, creating attributes and facts, building a logical data model, and other project design tasks, see the Project Design Guide.

The project life cycle

A MicroStrategy business intelligence application consists of many objects within projects. These objects are ultimately used to create reports that display data to the end user. As in other software systems, these objects should be developed and tested before they can be used in a production system. We call this process the project life cycle. This section discusses several project life cycle scenarios and the tools you can use to implement them.

In many cases, an application consists of a single project delivered to an end user. MicroStrategy OEM developers may choose to bundle several projects together to make a single application.

- For a description of the recommended scenario, see *Recommended scenario: Development, test, and production, page 220*
- For a real-life scenario, see *Real-life scenario: New version from a project developer, page 222*
- For details on how to implement the project life cycle in your MicroStrategy environment, see *Implementing the recommended life cycle, page 223*

Recommended scenario: Development, test, and production

This commonly used scenario is the project life cycle that MicroStrategy recommends you use as you develop your projects. In this scenario, you typically use three environments: development, test, and production. Each environment contains a MicroStrategy project.

MicroStrategy recommends that if you want to copy objects between two projects, such as from the development project to the test project, those projects should be related. Two projects are considered to be related if one was originally a duplicate of the other. To establish different development, test, and production projects, for example, you can create the test project by copying the development project, and you can create the production project by copying the test project. All three of these projects are related to each other. For more information about duplicating a project, see *Duplicating a project, page 225*.

 © 2016, MicroStrategy Inc.

This scenario is shown in the diagram below in which objects iterate between the development and test projects until they are ready for general users. Once ready, they are promoted to the production project.

The development project

In the development environment project, you create objects. This may be a project in which developers work. They think about the design of the whole system as they create the project's schema and application objects. For detailed instructions on how to design a project schema and create application objects, see the Project Design Guide.

The test project

Once the objects' definitions have stabilized, you move them to a test project that a wider set of people can use for testing. You may have people run through scripts or typical usage scenarios that users at your organization commonly perform. The testers look for accuracy (are the numbers in the reports correct?), stability (did the objects work? do their dependent objects work?), and performance (did the objects work efficiently, not producing overload on the data warehouse?).

In this test environment, you want the project to initially connect to a development data warehouse for initial testing. Later, for more stringent testing, connect the test project to the production data warehouse. If objects need further work, they are changed in the development project and recopied to the test project, but not changed in the test project.

The production project

After the objects have been tested and shown to be ready for use in a system accessible to all users, you copy them into the production project. This is the project used by most of the people in your company. It provides up-to-date reports and tracks various business objectives.

Implementing the recommended scenario

When migrating changes into a testing or development environment, be as thorough as possible. Carefully consider how your business users will access and use their application, reports, and dashboards on a daily basis. Anticipate the needs of your business users, and test every type of scenario before officially migrating to a production environment.

To set up the development, test, and production projects so that they all have related schemas, you need to first create the development project. For instructions on how to create a project, see the *MicroStrategy Project Design Guide*. Once the development project has been created, you can duplicate it to create the test and production projects

© 2016, MicroStrategy Inc.

using the Project Duplication Wizard. For detailed information about the Project Duplication Wizard, see *Duplicating a project, page 225*.

Once the projects have been created, you can migrate specific objects between them via Object Manager. For example, after a new metric has been created in the development project, you can copy it to the test project. For detailed information about Object Manager, see *Copying objects between projects: Object Manager, page 232*.

You can also merge two related projects with the Project Merge Wizard. This is useful when you have a large number of objects to copy. The Project Merge Wizard copies all the objects in a given project to another project. For an example of a situation in which you would want to use the Project Merge Wizard, see *Real-life scenario: New version from a project developer, page 222*. For detailed information about Project Merge, see *Merging projects to synchronize objects, page 266*.

To help you decide whether you should use Object Manager or Project merge, see *Comparing Project Merge to Object Manager, page 231*.

The Project Comparison Wizard can help you determine what objects in a project have changed since your last update. You can also save the results of search objects and use those searches to track the changes in your projects. For detailed information about the Project Comparison Wizard, see *Comparing and tracking projects, page 272*. For instructions on how to use search objects to track changes in a project, see *Tracking your projects with the Search Export feature, page 274*.

Integrity Manager helps you ensure that your changes have not caused any problems with your reports. Integrity Manager executes some or all of the reports in a project, and can compare them against another project or a previously established baseline. For detailed information about Integrity Manager, see *Chapter 16, Verifying Reports and Documents with Integrity Manager*.

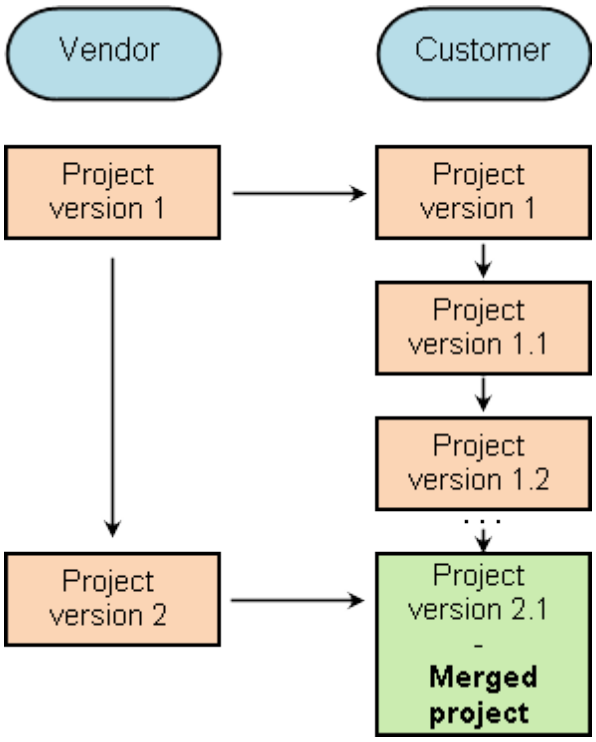
Real-life scenario: New version from a project developer

In this scenario, you have initially purchased a project from a vendor whose products are specialized for analyzing sales data. This is project version 1. Over the course of time, your developers have customized objects in the project, resulting in what you called version 1.1 and later, version 1.2., and so on. Now you have purchased version 2 of the project from the same vendor, and you wish to merge the new (Version 2) project with your existing (Version 1.2) project.

MicroStrategy encourages vendors in these situations to include in the installation of version 2 an “automatic” upgrade to the project using Project Merge. In this way the vendor, rather than the user or purchaser, can configure the rules for this project merge. For information about executing Project Merge without user input, see *Running Project Merge from the command line, page 268*.

This combination of the two projects creates Project version 2.1, as shown in the diagram below.

 © 2016, MicroStrategy Inc.



The vendor's new Version 2 project has new objects that are not in yours, which you feel confident in moving over. But some of the objects in the Version 2 project may conflict with objects that you had customized in the Version 1.2 project. How do you determine which of the Version 2 objects you want move into your system, or which of your Version 1.2 objects to modify?

You could perform this merge object-by-object and migrate them manually using Object Manager, but this will be time-consuming if the project is large. It may be more efficient to use the Project Merge tool. With this tool, you can define rules for merging projects that help you identify conflicting objects and handle them a certain way. Project Merge then applies those rules while merging the projects. For more information about using the MicroStrategy Project Merge tool, see *Merging projects to synchronize objects, page 266*.

Implementing the recommended life cycle

The following section provides a high-level, simplified overview of the procedure for implementing the recommended project life cycle in your company's MicroStrategy environment. This is a simplified version of the workflow you are likely to see at your organization. However, you should be able to apply the basic principles to your specific situation.

1 *Create the development project.*

Creating the development project involves setting up the database connections and project schema, configuring user security, and building the initial schema and application objects. For information on creating a project, see the *MicroStrategy Project Design Guide*.

2 *Create the test and production projects by duplicating the development project.*

MicroStrategy recommends that you duplicate the development project to create the test and production projects, rather than creating them separately. Duplicating ensures that all three projects have related schemas, enabling you to safely use Object Manager or Project Merge to copy objects between the projects.

For instructions on how to duplicate a project, see *Duplicating a project, page 225*.

3 *Create objects in the development project.*

In the recommended scenario, all objects (attributes, metrics, reports) are created in the development project, and then migrated to the other projects. For more information about the development project, see *The development project, page 221*.

For instructions on creating schema objects, see the *MicroStrategy Project Design Guide*. For instructions on creating application objects, see the *MicroStrategy Basic Reporting Guide* and *MicroStrategy Advanced Reporting Guide*.

4 *Migrate objects from the development project to the test project.*

Once the objects have been created and are relatively stable, they can be migrated to the test project for testing. For instructions on how to migrate objects, see *Updating projects with new objects, page 230*.

Depending on the number of objects you have created or changed, you can use either Object Manager or Project Merge to copy the objects from the development project to the test project. For a comparison of the two tools, see *Comparing Project Merge to Object Manager, page 231*. For a tool to determine what objects have changed, see *Comparing and tracking projects, page 272*.

5 *Test the new objects.*

Testing involves making sure that the new objects produce the expected results, do not cause data errors, and do not put undue strain on the data warehouse. If the objects are found to contain errors, these errors are reported to the development team so that they can be fixed and tested again. For more information about the test project, see *The test project, page 221*.

Integrity Manager is an invaluable tool in testing whether new objects cause reports to generate different results. For detailed information about Integrity Manager, see *Chapter 16, Verifying Reports and Documents with Integrity Manager*.

6 *Migrate objects from the test project to the production project.*

Once the objects have been thoroughly tested, they can be migrated to the production project and put into full use. For instructions on how to migrate objects, see *Updating projects with new objects, page 230*.

7 *Repeat steps 3 through 6 as necessary.*

 © 2016, MicroStrategy Inc.

The project life cycle does not end with the first migration of new objects into the production project. A developer may come up with a new way to use an attribute in a metric, or a manager may request a specific new report. These objects pass through the project life cycle in the same way as the project's initial objects.

Duplicating a project

Duplicating a project is an important part of the application life cycle. If you want to copy objects between two projects, MicroStrategy recommends that the projects have related schemas. This means that one must have originally been a duplicate of the other, or both must have been duplicates of a third project.

Autostyles, which give a uniform appearance to reports, can be freely moved between projects regardless of whether their schemas are related. For instructions on migrating autostyles between projects, see the Advanced Reporting Guide.

Project duplication is done using the Project Duplication Wizard. For detailed information about the duplication process, including step-by-step instructions, see *The Project Duplication Wizard, page 227*.

To migrate a project from a Microsoft Access database to another database platform, you must use the Project Mover Wizard. For detailed information about this migration, see *Migrating a project to a new database platform, page 228*.

You can duplicate a MicroStrategy project in one of the following ways:

- From a three-tier (server) project source to a three-tier (server) project source
- From a three-tier (server) project source to a two-tier (direct) project source
- From a two-tier (direct) project source to a two-tier (direct) project source
- From a two-tier (direct) project source to a three-tier (server) project source

A server (three-tier) project source is connected to an Intelligence Server, and has the full range of administrative options available. A direct (two-tier) project source is not connected to an Intelligence Server. For more information on three-tier and two-tier project sources, see the Project Design Guide.

Do not refresh the warehouse catalog in the destination project. Refresh the warehouse catalog in the source project, and then use Object Manager to move the updated objects into the destination project. For information about the warehouse catalog, see the *Optimizing and Maintaining your Project* chapter in the Project Design Guide.

What objects are duplicated with a project?

When you duplicate a project, all schema objects (attributes, facts, hierarchies, and transformations) are duplicated. By default all application objects (reports, documents, metrics, and so forth) contained in the project are also duplicated.

If you are copying a project to another project source, you have the option to duplicate configuration objects as well. Specifically:

- You can choose whether to duplicate all configuration objects, or only the objects used by the project.
- You can choose to duplicate all users and groups, only the users and groups used by the project, no users and groups, or a custom selection of users and groups.
- You can choose to duplicate user, contact, and subscription information.

For each type of configuration object (user/group, security role, schedule, contact/contact group, database connection/instance, database login) you must choose whether to duplicate the object if it already exists in the destination project source metadata. For users/groups and security roles, you can also choose to merge the privileges of the source and destination versions.

Duplicating projects in multiple languages

When you duplicate a project that contains warehouse data in multiple languages, you have the option of duplicating all, some, or none of those languages. In addition, you can select the new default language for the project.

Whenever you duplicate a project or update the metadata, a language check ensures that the language settings in the CURRENT_USER registry key, the LOCAL_MACHINE registry key, and the Project locale property all match before an update takes place. The system performs the following checks:

- In a direct (two-tier) configuration, without an Intelligence Server, the system checks that the language under the LOCAL_MACHINE registry key matches the language under the CURRENT_USER registry key.
- In a server (three-tier) configuration, with an Intelligence Server, the system checks that the language under the CURRENT_USER registry key on the client machine matches the language under the LOCAL_MACHINE registry key on the server machine.

The MicroStrategy interface obtains the language information from the CURRENT_USER registry key and the server obtains the language information from the LOCAL_MACHINE registry key. This can lead to inconsistencies in the language display. The language check prevents these inconsistencies and ensures that the language display is consistent across the interface.

The internationalization settings in Object Manager allow you to create related projects in different languages. For more information on this process, see *Copying objects between projects in different languages, page 242*.

 © 2016, MicroStrategy Inc.

The Project Duplication Wizard

You should always use the Project Duplication Wizard to duplicate your projects. This ensures that all project objects are duplicated properly, and that the new project's schema is identical to the source project's schema.

To duplicate a project, you must have the Bypass All Object Security Access Checks privilege for that project. In addition, you must have the Create Schema Objects privilege for the target project source.

The following high-level procedure provides an overview of what the Project Duplication Wizard does. For an explanation of the information required at any given page in the wizard, see the Help (from the wizard, click **Help**, or press **F1**).

High-level steps to duplicate a project with the Project Duplication Wizard

1. From Object Manager select the **Project** menu (or from Developer select the **Schema** menu), then select **Duplicate Project**. The Project Duplication Wizard opens.
2. Specify the project source and project information that you are copying from (the source).
3. Specify the project source and project information that you are copying to (the destination).
4. Indicate what types of objects to copy.
5. Specify whether to keep or merge configuration object properties if these already exist in the destination project source. For example, if properties such as password expiration and so on are different by default between the project sources, which set of properties do you want to use?
6. Specify whether you wish to see the event messages as they happen and, if so, what types. Also specify whether to create log files and, if so, what types of events to log, and where to locate the log files. By default Project Duplicator shows you error messages as they occur, and logs most events to a text file. This log file is created by default in `C:\Program Files (x86)\Common Files\MicroStrategy\`.

Scheduling project duplication

At the end of the Project Duplication Wizard, you are given the option of saving your settings in an XML file. You can load the settings from this file later to speed up the project duplication process. The settings can be loaded at the beginning of the Project Duplication Wizard.

You can also use the settings file to run the wizard in command-line mode. The Project Duplication Wizard command line interface enables you to duplicate a project without having to load the graphical interface, or to schedule a duplication to run at a specific time. For example, you may want to run the project duplication in the evening, when the

© 2016, MicroStrategy Inc.

load on Intelligence Server is lessened. You can create an XML settings file, and then use the Windows `AT` command or the Unix scheduler to schedule the duplication to take place at night.

To duplicate a project from the command line

After saving the settings from the Project Duplication Wizard, invoke the Project Duplication Wizard executable `ProjectDuplicate.exe`. By default this executable is located in `C:\Program Files (x86)\Common Files\MicroStrategy`.

The syntax is:

```
ProjectDuplicate.exe -f Path\XMLFilename [-sp
SourcePassword] [-dp DestinationPassword] [-sup] [-md] [-
dn OverwriteName]
```

where:

- *`Path`* is the path to the saved XML settings file.
- *`XMLFilename`* is the name of the saved XML settings file.
- *`SourcePassword`* is the password for the source project's project source.
- *`TargetPassword`* is the password for the destination project's project source.
- `-sup` indicates that feedback messages will be suppressed (silent mode).
- `-md` indicates that the metadata of the destination project source will be updated if it is older than the source project source's metadata.
- `-dn` *`OverwriteName`* specifies the name of the destination project. This overrides the name specified in the XML settings file.

For information on the syntax for the Windows `AT` command or a UNIX scheduler, see the documentation for your operating system.

Migrating a project to a new database platform

The Project Mover Wizard guides you through the process of migrating a MicroStrategy project from a Microsoft Access database to a new database platform. This can be useful for creating and testing a new project before it is placed in the production database.

For example, a business analyst has an idea for a new business intelligence application using MicroStrategy. The analyst needs to create a proof-of-concept project to show her manager. The project will eventually be used in the development and production environment, but the system administrator might decide that it is not ideal to create the demo project in the production database. Instead the analyst puts the project together on her laptop, using a local Microsoft Access database. Once she demonstrates the project and receives approval for it, the administrator can use the Project Mover Wizard to move the project from the laptop's Access database into the development environment's database platform.

 © 2016, MicroStrategy Inc.

The following high-level procedure provides an overview of what the Project Mover Wizard does. For an explanation of the information required at any given page in the wizard, see the Help (from the wizard, click **Help**, or press **F1**).

Prerequisites

- The project must be contained in a Microsoft Access database.
- To migrate a project to a new database platform, you must have the Bypass All Object Security Access Checks privilege for that project.
- The source project and destination project source must be the same version of MicroStrategy metadata. For example, if you have updated the destination project source to the latest version of MicroStrategy, you must also update the source project before using Project Mover to migrate it. For information about updating a project's metadata, see the Upgrade Guide.

To migrate a project to a different database

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Project Mover**. The Project Mover Wizard opens.
2. Select the Microsoft Access warehouse and metadata databases that contain the source project, and then select the source project.
3. Select any SQL scripts you want to run on the data warehouse, either before or after project migration.
4. Select the database into which the project is to be migrated.
5. If project metadata already exists in the destination database, select whether to append the migrated project to the existing data, or overwrite that data.
6. Review your choices and click **Finish** on the Summary page of the Project Mover Wizard. The wizard migrates your project to the new database.

Migrating a project automatically

In addition to accepting user input and then immediately moving a project, the Project Mover Wizard can record your input in a response file. This file contains all the information needed to move a project to a new database platform, and can be later used to move a project without any additional user input. This is especially useful for scheduling a project migration for a time when there are not many users on the system.

To create a response file, from the first page of the Project Mover Wizard click **Advanced**. On the Advanced Options page, select **Generate a response file** and enter the name and location of the new response file in the text field.

To execute a response file from the Project Mover Wizard, from the first page of the wizard click **Advanced**. Then select the **Use Response File** option and load the response file. The Wizard opens the Summary page, which lists all the options set by the response

file. After reviewing these options, click **Finish**. The Project Mover Wizard begins moving the project.

To execute a response file from the command line, you need to invoke the Project Mover executable, `demomover.exe`. By default, this directory is `C:\Program Files (x86)\Common Files\Microstrategy`.

The syntax is:

```
demomover.exe -r "File Location\Filename.ini"
```

where "`File Location`" is the path to the response file and "`Filename.ini`" is the name of the response file.

Updating projects with new objects

When you create or modify an object in your development environment, you eventually need to copy that object to the test project, and later to the production project.

For example, a developer creates a new metric in the development project. Once the metric is ready to be tested, it needs to be present in the test project. You could re-create the metric in the test project based on the same specifications, but it can be easy to miss an important setting in the metric. A quicker and more reliable method is to use MicroStrategy Object Manager to migrate the new metric from the development project to the test project. Then, when the metric is ready to be rolled out to your users, you can use Object Manager again to migrate it from the test project to the production project.

MicroStrategy has the following tools available for updating the objects in a project:

- Object Manager migrates a few objects at a time. For information about Object Manager, see *Copying objects between projects: Object Manager, page 232*.
- An update package migrates a previously specified group of objects. Update packages are part of Object Manager. For information about update packages, see *Copying objects in a batch: Update packages, page 249*.
- Project Merge migrates all the objects in a project at once. For information about Project Merge, see *Merging projects to synchronize objects, page 266*.

For a comparison of these tools, see *Comparing Project Merge to Object Manager, page 231*.

Note the following:

- If you want to move or copy objects between projects, MicroStrategy recommends that those projects have related schemas. This means that either one project must be a duplicate of the other, or both projects must be duplicates of a third project. For information about duplicating projects, including instructions, see *Duplicating a project, page 225*.

 © 2016, MicroStrategy Inc.

- If one of the projects is updated to a new MicroStrategy release, but another project is not updated, you cannot move or copy objects between the projects. You must first update the other project before you can copy objects between the projects.

Comparing Project Merge to Object Manager

Object Manager and Project Merge are both designed for migrating objects between projects. Both tools involve copying objects between projects in a definite order according to object types. Which tool you should use depends on several factors, such as how many objects you need to move at once. The following are some of the differences between the tools:

- Object Manager can move just a few objects, or just the objects in a few folders. Project Merge moves all the objects in a project.
- Using Object Manager to merge whole projects means moving many objects individually or as a subset of all objects. This can be a long and tedious task. Project Merge packages the functionality for easier use because it moves all objects at one time.
- Object Manager must locate the dependents of the copied objects and then determine their differences before performing the copy operation. Project Merge does not do a dependency search, since all the objects in the project are to be copied.
- The Project Merge Wizard allows you to store merge settings and rules in an XML file. These rules define what is copied and how conflicts are resolved. Once they are in the XML file, you can load the rules and "replay" them with Project Merge. This can be useful if you need to perform the same merge on a recurring schedule. For example, if a project developer sends you a new project version quarterly, Project Merge can make this process easier.
- Project Merge can be run from the command prompt in Microsoft Windows. An added benefit of this feature is that project merges can be scheduled using the `at` command in Windows and can be run silently in an installation routine.
- The changes to be made through Object Manager can be saved as an update package and applied at a later time. For instructions on how to create and use update packages, see *Copying objects in a batch: Update packages, page 249*.
- The changes to be made through an Object Manager update package can be reversed using an undo package. For instructions on how to roll back changes using Object Manager, see *Rolling back changes: Undo packages, page 265*.

Locking projects

When you open a project in Project Merge, you automatically place a metadata lock on the project. You also place a metadata lock on the project if you open it in read/write mode in Object Manager, or if you create or import an update package from the command line. For more information about read/write mode versus read-only mode in Object Manager, see *Project locking with Object Manager, page 233*.

© 2016, MicroStrategy Inc.

A metadata lock prevents other MicroStrategy users from modifying any objects in the project in Developer or MicroStrategy Web, while objects are being copied with Object Manager or Project Merge. It also prevents other MicroStrategy users from modifying any configuration objects, such as users or groups, in the project source. Locking a project prevents metadata inconsistencies.

When other users attempt to open an object in a locked project using Developer or MicroStrategy Web, they see a message that informs them that the project is locked because a user that opened the project first is modifying it. Users can then choose to open the object in read-only mode or view more details about the lock. Users can execute reports in a locked project, but the report definition that is used is the last definition saved prior to the project being locked.

If you lock a project by opening it in Object Manager, you can unlock the project by right-clicking the project in Object Manager, and choosing **Disconnect from Project Source**.

Only the user who locked a project, or another user with the Bypass All Object Security Access Checks and Create Configuration Objects privileges, can unlock a project.

You can also lock or unlock a project or a configuration manually using Developer. For detailed steps on locking and unlocking projects manually, see the *MicroStrategy Developer Help*.

Command Manager scripts can be used to automate metadata lock management. For information about Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*. For Command Manager syntax for managing metadata locks, see the Command Manager Help (press **F1** from within Command Manager).

Copying objects between projects: Object Manager

MicroStrategy Object Manager can help you manage objects as they progress through your project's life cycle. Using Object Manager, you can copy objects within a project or across projects.

Object Manager and Project Merge both copy multiple objects between projects. Use Object Manager when you have only a few objects that need to be copied. For the differences between Object Manager and Project Merge, see *Comparing Project Merge to Object Manager, page 231*.

This section includes:

- *Prerequisites for copying objects between projects, page 233*
- *Project locking with Object Manager, page 233*
- *Copying objects, page 234*

 © 2016, MicroStrategy Inc.

- *What happens when you copy or move an object, page 237*
- *Resolving conflicts when copying objects, page 243*
- *Copying objects in a batch: Update packages, page 249*

Prerequisites for copying objects between projects

- To use Object Manager to copy objects between projects, you must have the Use Object Manager privilege for both projects. You do not need to have ACL permissions for the objects you are migrating, because the Use Object Manager privilege automatically grants its user the Bypass All Object Security Access Checks privilege when that user is using Object Manager.
- To create an update package, you must have either the Use Object Manager privilege or the Use Object Manager Read-only privilege for the project from which you are creating an update package.
- If you want to migrate objects between projects with Object Manager, MicroStrategy recommends that those projects have related schemas. This means that either one project must be a duplicate of the other, or both projects must be duplicates of a third project. For information about duplicating projects, including instructions, see *Duplicating a project, page 225*.
- To move system objects between projects that do not have related schemas, the projects must either have been created with MicroStrategy 9.0.1 or later, or have been updated to version 9.0.1 or later using the **Perform system object ID unification** option. For information about this upgrade, see the Upgrade Guide.
- If one of the projects is updated to a new MicroStrategy release, but another project is not updated, you cannot move or copy objects from the project using the updated version of MicroStrategy to the older version. However, you can move objects from the older version to the updated project if the older version is interoperable with the updated version. For detailed information about interoperability between versions of MicroStrategy, see the *MicroStrategy Readme*.

Project locking with Object Manager

Opening a connection to a project with Object Manager causes the project metadata to become locked. Other users cannot make any changes to the project until it becomes unlocked. For detailed information about the effects of locking a project, see *Locking projects, page 231*.

If you need to allow other users to change objects in projects while the projects are opened in Object Manager, you can configure Object Manager to connect to projects in read-only mode. You can also allow changes to configuration objects by connecting to project sources in read-only mode.

Connecting to a project or project source in read-only mode has the following limitations:

- A connection in read-only mode may not display the most recent information. For example, if you view a folder in Object Manager in a read-only connection, and then another user adds an object to that folder, the object is not displayed in Object Manager.
- You cannot copy objects into a read-only project or project source. If you connect to a project in read-only mode, you can still move, copy, and delete objects in a project, but you cannot copy objects from another project into that project.
- By default, users cannot create update packages in read-only mode. This is because objects, and their used dependencies, may be changed between the time they are selected for inclusion in the update package and the time the package is actually generated. If necessary, you can configure Object Manager to allow the creation of update packages in read-only mode. For information about update packages, see *Copying objects in a batch: Update packages, page 249*.

To open projects or connections in read-only mode

1. From the **Tools** menu, select **Preferences**. The Object Manager Preferences dialog box opens.
2. Expand the **Object Manager** category, and then select **Connection**.
3. To open project sources in read-only mode, select the **Open configuration in read-only mode** check box.
4. To open projects in read-only mode, select the **Open project in read-only mode** check box.
5. To allow the creation of update packages in read-only mode, select the **Allow update package creation in read-only mode** check box.
6. Click **OK**. The Object Manager Preferences dialog box closes and your preferences are saved.

Copying objects

Object Manager can copy application, schema, and configuration objects.

- **Application objects** include reports and documents, and the objects used to create them, such as templates, metrics, filters, prompts, and searches. Folders are also considered to be application objects.
- **Schema objects** include attributes, facts, hierarchies, transformations, functions, partition mappings, columns, and tables.
- **Configuration objects** include objects that are used by all projects in a project source, such as users and user groups, database instances and logins, security roles, and Distribution Services devices, transmitters, and contacts.

 © 2016, MicroStrategy Inc.

If you use Object Manager to copy a user or user group between project sources, the user or group reverts to default inherited access for all projects in the project source. To copy a user or group's security information for a project, you must copy the user or group in a configuration update package. For information about update packages, see *About update packages, page 250*.

For background information on these objects, including how they are created and what roles they perform in a project, see the Project Design Guide.

In a MicroStrategy system, each object has a unique Object ID. Object Manager identifies objects based on their Object ID, not their name. Hence, objects with different names are treated as versions of the same object if they have the same Object ID.

Best practices for copying objects

MicroStrategy recommends that you observe the following practices when copying objects:

- Back up your metadata before copying any objects. Object Manager cannot undo the copying and replacing of objects.
- Ensure that the Dependency Search, Conflict Resolution, International, and Migration options in the Object Manager Preferences dialog box are set to fit your project's needs. For details about the Dependency Search options, see *Migrating dependent objects, page 240*. For details about the Conflict Resolution options, see *Resolving conflicts when copying objects, page 243*. For details about the Migration options, see *What happens when you copy or move an object, page 237*. The Object Manager Help also provides a detailed explanation for each of these options.
- Copy application objects into the following project folders:
  - My Personal Objects or any subfolder of My Personal Objects
  - Public Objects or any subfolder of Public Objects.
- Copy schema objects into the appropriate Schema Objects sub- or descendent folders only. For example, if you are copying a hierarchy, you should only paste the hierarchy into the *Project Name*\Schema Objects\Hierarchies folder.
- When copying MDX cubes between projects, make sure that the conflict resolution action for the cubes, cube attributes, and reports that use the cubes is set to Replace.
- If you need to copy objects from multiple folders at once, you can create a new folder, and create shortcuts in the folder to all the objects you want to copy. Then copy that folder. Object Manager copies the folder, its contents (the shortcuts), and their dependencies (the target objects of those shortcuts) to the new project.
- Another way to copy objects from multiple folders at once is to create an update package from the source project, and then import it into the target project. For more information about update packages, including step-by-step instructions, see *Copying objects in a batch: Update packages, page 249*.
- If you are using update packages to update the objects in your projects, use the Export option to create a list of all the objects in each update package.

© 2016, MicroStrategy Inc.

- When copying objects that contain location-specific strings (such as metric aliases, custom group names, or text boxes in documents), make sure that you either disable Advanced Conflict Resolution, or use the same option in the translation preferences and in the conflict resolution. Otherwise there may be inconsistencies between the object definition and the translation in the destination project. For an explanation of the advanced conflict resolution options, including how to enable or disable these options, see *Copying objects between projects in different languages, page 242*.
- Regardless of the translation preferences, when copying objects with location-specific strings, you should always verify the results. For example, empty translations in the source or destination may result in incorrect translations being saved with the new object. You can use Integrity Manager to identify reports or documents that have unexpected translations. For information about Integrity Manager, see *Chapter 16, Verifying Reports and Documents with Integrity Manager*.

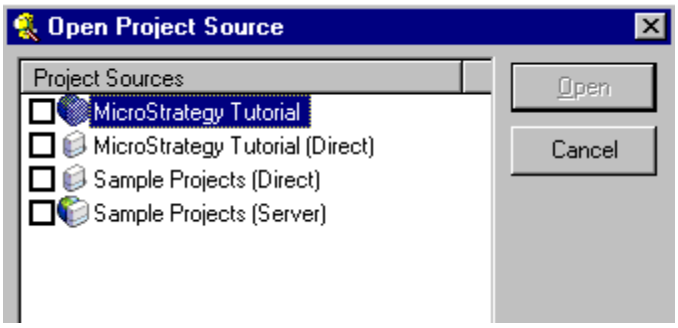
To copy objects between projects

Note the following:

- To log in to a project source using Object Manager, you must have the Use Object Manager privilege for that project.
- If you want to copy application or schema objects between projects, MicroStrategy recommends that the two projects have related schemas (one must be a duplicate of the other or both must be duplicates of a common project). For details on this, see *Duplicating a project, page 225*.

Log in to the projects in Object Manager

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **Object Manager**. The Open Project Source dialog box opens.



2. In the list of project sources, select the check box for the project source you want to access. You can select more than one project source.
3. Click **Open**. You are prompted to log in to each project source that you have selected.
4. When you have logged into each project source, MicroStrategy Object Manager opens.

 © 2016, MicroStrategy Inc.

Use the appropriate sub-procedure below depending on whether you want to *Copy application and schema objects, page 237* or *Copy configuration objects, page 237*.

Copy application and schema objects

5 In the Folder List, expand the project that contains the object you want to copy, then navigate to the object.

6 Copy the object by right-clicking and selecting **Copy**.

7 Expand the destination project in which you want to paste the object, and then select the folder in which you want to paste the object.

8 Paste the application or schema object into the appropriate destination folder by right-clicking and selecting **Paste**.

For information about additional objects that may be copied with a given object, see *Used dependencies, page 238*.

If you are copying objects between two different project sources, two windows are open within the main Object Manager window. In this case, instead of right-clicking and selecting **Copy** and **Paste**, you can drag and drop objects between the projects.

9 If you copied any schema objects, you must update the destination project's schema. Select the destination project, and from the **Project** menu, select **Update Schema**.

Copy configuration objects

10 In the Folder Lists for both the source and destination projects, expand the **Administration** folder, then select the appropriate manager for the type of configuration object you want to copy (Database Instance Manager, Schedule Manager, or User Manager).

11 From the list of objects displayed on the right-hand side in the source project source, drag the desired object into the destination project source and drop it.

To display the list of users on the right-hand side, expand User Manager, then on the left-hand side select a group.

What happens when you copy or move an object

If the object you are copying does not exist in the destination project, MicroStrategy Object Manager copies the object into the destination project. This new object has the same name as the source object.

If the object you are copying does exist in the destination project, a conflict occurs and Object Manager opens the Conflict Resolution dialog box. For information about how to resolve conflicts, see *Resolving conflicts when copying objects, page 243*.

For more information about handling specific situations, see:

© 2016, MicroStrategy Inc.

- *Managing object dependencies, page 238*
- *Migrating dependent objects, page 240*
- *Timestamps for migrated objects, page 242*
- *Copying objects between projects in different languages, page 242*

Managing object dependencies

When an object uses another object in its definition, the objects are said to depend on one another. Object Manager recognizes two types of object dependencies: used dependencies and used-by dependencies.

When you migrate an object to another project, by default any objects used by that object in its definition (its used dependencies) are also migrated. You can exclude certain objects and tables from the dependency check and migration. For instructions, see *Excluding dependent attributes or tables from object migration, page 241*.

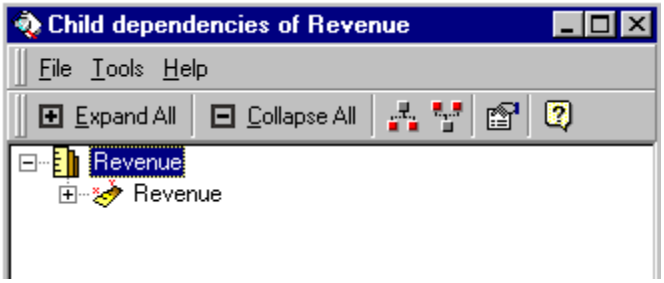
Used dependencies

A used dependency occurs when an object uses other objects in its definition. For example, in the MicroStrategy Tutorial project, the metric named Revenue uses the base formula named Revenue in its definition. The Revenue metric is said to have a used dependency on the Revenue base formula. (Additionally, the Revenue base formula has a used-by dependency of the Revenue metric.)

When you migrate an object to another project, any objects used by that object in its definition (its used dependencies) are also migrated. The order of these dependent relationships is maintained.

To manage used or used-by dependencies of an object

1. After you have opened a project source and a project using Object Manager, in the **Folder List** select the object.
2. From the **Tools** menu, select **Object used dependencies**. The Used dependencies dialog box opens and displays a list of objects that the selected object uses in its definition. The image below shows the used dependencies of the Revenue metric in the MicroStrategy Tutorial project: in this case, the used dependency is the Revenue base formula.



 © 2016, MicroStrategy Inc.

3 In the Used dependencies dialog box, you can do any the following:

- View used dependencies for any object in the list by selecting the object and clicking the **Object used dependencies** toolbar icon.
- Open the Used-by dependencies dialog box for any object in the list by selecting the object and clicking the **Object used-by dependencies** icon on the toolbar. For information about used-by dependencies, see *Used-by dependencies, page 239*.
- View the properties of any object, such as its ID, version number, and access control lists, by selecting the object and from the **File** menu choosing **Properties**.

Used-by dependencies

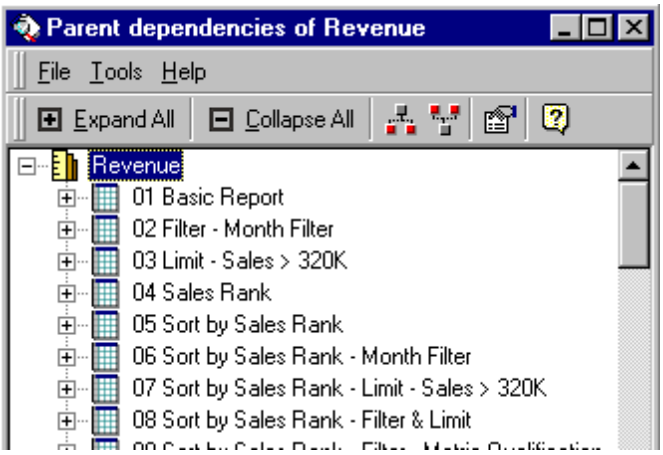
A used-by dependency occurs when an object is used as part of the definition of other objects. For example, in the MicroStrategy Tutorial project, the Revenue metric has used-by dependencies of many reports and even other metrics. The Revenue metric is said to be used by these other objects.

Used-by dependents are not automatically migrated with their used objects. However, you cannot delete an object that has used-by dependencies without first deleting its used objects.

To manage the used-by dependencies of an object

1 After you have opened a project source and a project using Object Manager, from the **Folder List** select the object.

2 From the **Tools** menu, choose **Object used-by dependencies**. The Used-by dependencies dialog box opens and displays a list of objects that depend on the selected object for part of their definition. The image below shows some of the used-by dependencies for the Revenue metric in the MicroStrategy Tutorial project.



3 In the Used-by dependencies dialog box, you can do any of the following:

- View used-by dependencies for any object in the list by selecting the object and clicking the **Object used-by dependencies** icon on the toolbar.
- Open the Used dependencies dialog box for any object in the list by selecting the object and clicking the **Object used dependencies** icon on the toolbar. For information about used dependencies, see *Used dependencies, page 238*.
- View the properties of any object, such as its ID, version number, and access control lists, by selecting the object and from the **File** menu choosing **Properties**.

Migrating dependent objects

When you copy an object using Object Manager, it checks for any used dependents of that object and copies them as well. These dependent objects are copied to the same path as in the source project. If this path does not already exist in the destination project, Object Manager creates the path.

For example, a user copies a report from the source project to the destination project. In the source project, all dependents of the report are stored in the `Public Objects\Report Dependents` folder. Object Manager looks in the destination project's Public Objects folder for a subfolder named `Report Dependents` (the same path as in the source project). If the folder exists, the dependent objects are saved in that folder. If the destination project does not have a folder in Public Objects with the name `User`, Object Manager creates it and saves all dependent objects there.

When you create an update package, click **Add All Used Dependencies** to make sure all used dependencies are included in the package. If the dependent objects for a specific object do not exist in either the destination project source or in the update package, the update package cannot be applied. If you choose not to add dependent objects to the package, make sure that all dependent objects are included in the destination project source.

Object dependencies

Some objects have dependencies that are not immediately obvious. These are listed below:

- **Folders** have a used dependency on each object in the folder. If you copy a folder using Object Manager, all the objects in that folder are also copied.

A folder that is copied as part of an update package does not have a used dependency on its contents.

- **Shortcut objects** have a used dependency on the object they are a shortcut to. If you copy a shortcut using Object Manager, the object it is a shortcut to is also copied.
- **Security filters**, **users**, and **user groups** have a used dependency on the user groups they belong to. If you copy a security filter, user, or user group, the groups that it belongs to are also copied.

 © 2016, MicroStrategy Inc.

Groups have a used-by dependency on the users and security filters that are associated with them. Copying a group does not automatically copy the users or security filters that belong to that group. To copy the users or security filters in a group, select the users from a list of that group's used-by dependents and then copy them.

- **Attributes used in fact expressions** are listed as dependents of the fact. When the fact is copied, the attribute is also copied.

Attributes used in fact entry levels are not dependents of the fact.

Excluding dependent attributes or tables from object migration

When you copy an object, or add dependent objects to an update package, Object Manager searches for that object's used dependencies so it can copy those objects also. Depending on the options you set in the Object Manager Preferences, you can exclude certain types of dependent objects from this migration.

The options are:

- **Exclude all parent attributes from an attribute** and **Exclude all child attributes from an attribute**: An attribute has a used dependency on its parent and child attributes in a hierarchy. Thus, migrating an attribute may result in migrating its entire hierarchy. To exclude the parent or child attributes from being migrated, select the corresponding option.
- **Exclude non-lookup tables from an attribute** and **Exclude all tables from a fact**: An attribute or fact has a used dependency on each table that is referenced by the attribute or fact. Thus, by default, migrating an attribute or fact results in migrating all its associated tables. You can choose to exclude the tables from the dependency search if, for example, you have mapped additional tables to an attribute or fact for testing purposes but do not need those tables in the production project.

For attributes, the lookup table must always exist in the destination project, so it is always migrated.

To exclude types of dependent objects

1. From the **Tools** menu, select **Object Manager Preferences**. The Object Manager Preferences dialog box opens.
2. Expand **Dependency search**, and then select **Dependency search**.
3. Select the check boxes for the objects you want to exclude from Object Manager's dependency checking.
4. Click **OK**. The Object Manager Preferences dialog box closes and your preferences are saved.

Timestamps for migrated objects

By default, when an object is migrated, the object's modification timestamp is updated to the destination Intelligence Server's migration process time. You can change this behavior so that the timestamp remains as the last modification time the object had in the source project.

To set the migrated object modification timestamp

1. From the **Tools** menu, select **Object Manager Preferences**. The Object Manager Preferences dialog box opens.
2. Expand **Migration**, and then select **Migration**.
3. To cause objects to keep the modification timestamp from the source project, select the **Preserve object modification timestamp during migration** check box. If this check box is cleared, objects take the modification timestamp from the destination Intelligence Server at the time of migration.
4. Click **OK**. The Object Manager Preferences dialog box closes and your preferences are saved.

Copying objects between projects in different languages

Object Manager's internationalization options allow you to specify the locale settings to be used when copying objects. You can also retain the object's name, description, and long description from the destination project, when replacing objects in the destination project using Object Manager.

The ability to retain the name, description, and long description is important in internationalized environments. When replacing the objects to resolve conflicts, retaining these properties of the objects in the destination project facilitates support of internationalized environments. For example, if the destination project contains objects with French names but the source project has been developed in English (including English names), you can retain the French names and descriptions for objects in the destination project. Alternately, you can update the project with the English names and not change the object itself.

To set the internationalization options

1. From the **Tools** menu, select **Object Manager Preferences**. The Object Manager Preferences dialog box opens.
2. Expand the **International** category, and select **Language**.
3. From the **Interface Language** drop-down list, select the language to be used in Object Manager. By default this is the language used in all MicroStrategy products installed on this system.

 © 2016, MicroStrategy Inc.

4 From the **Language for metadata and warehouse data if user and project level preferences are set to default** drop-down list, select whether copied objects use the locale settings from Developer or from the machine's regional settings.

For more information on metadata and warehouse data languages, see *About internationalization* in the Supplemental Reference for System Administration, and for a table on the prioritization of user- and project-level language preferences, see *Configuring metadata object and report data language preferences* in the Supplemental Reference for System Administration.

5 In the **International** category, select **Translation**.

6 To resolve translations with a different action than that specified for the object associated with the translation, select the **Enable advanced conflict resolution** check box.

- To always use the translations in the destination project, select **Keep Existing**.
- To always use the translations in the source project, select **Replace**.

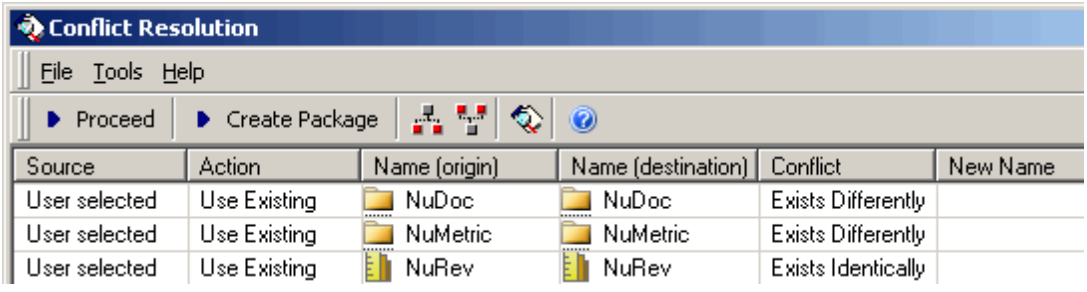
7 Select the **Merge translations even if object exists identically** check box to update the translations for all copied objects in the destination project, according to the option specified above (Keep Existing or Replace (Default)), even if the object exists identically in both projects.

8 Click **OK**. The Object Manager Preferences dialog box closes and your preferences are saved.

Resolving conflicts when copying objects

In the MicroStrategy system, every object has an ID (or GUID) and a version. The version changes every time the object is updated; the ID is created when the object is created and remains constant for the life of the object. To see the ID and version of an object, right-click the object and select **Properties**.

When copying objects across projects with Object Manager, if an object with the same ID as the source object exists anywhere in the destination project, a conflict occurs and the Conflict Resolution dialog box (shown below) opens. It prompts you to resolve the conflict.



The table below lists the different kinds of conflict:

Conflict	Explanation
Exists identically	The object ID, object version, and path are the same in the source and destination projects.
Exists differently	The object ID is the same in the source and destination projects, but the object versions are different. The path may be the same or different.
Exists identically except for path	The object ID and object version are the same in the source and destination projects, but the paths are different. This occurs when one of the objects exists in a different folder. **Note**: If your language preferences for the source and destination projects are different, objects that are identical between the projects may be reported as Exists Identically Except For Path. This occurs because when different languages are used for the path names, Object Manager treats them as different paths. To resolve this, set your language preferences for the projects to the same language. If you resolve the conflict with the **Replace** action, the destination object is updated to reflect the path of the source object.
Exists identically except for Distribution Services objects	(User only) The object ID and object version of the user are the same in the source and destination projects, but at least one associated Distribution Services contact or contact group is different. This may occur if you modified a contact or contact group linked to this user in the source project. If you resolve the conflict with the **Replace** action, the destination user is updated to reflect the contacts and contact groups of the source user.
Does not exist	The object exists in the source project but not in the destination project. **Note**: If you clear the **Show new objects that exist only in the source** check box in the **Migration** category of the Object Manager Preferences dialog box, objects that do not exist in the destination project are copied automatically with no need for conflict resolution.

Choosing an action to resolve a conflict

If a conflict occurs you must determine what action Object Manager should take. The different actions are explained in the table below.

When Object Manager reports a conflict it also suggests a default action to take for that conflict. For information on changing the default action, see *Setting default actions for conflict resolutions, page 246*.

User Action	Effect
Use existing	No change is made to the destination object. The source object is not copied.

 © 2016, MicroStrategy Inc.

User Action	Effect
Replace	The destination object is replaced with the source object. **Note**: If the conflict type is Exists Identically Except For Path, or Exists Identically Except For Distribution Services Objects, the destination object is updated to reflect the path or Distribution Services addresses and contacts of the source object. Replace moves the object into same parent folder as the source object. If the parent path is the same between source and destination but the grandparent path is different, Replace may appear to do nothing because Replace puts the object into the same parent path. Non-empty folders in the destination location will never have the same version ID and modification time as the source, because the folder is copied first and the objects are added to it, thus changing the version ID and modification times during the copy process.
Keep both	No change is made to the destination object. The source object is duplicated in the destination location.
Use newer	If the source object's modification time is more recent than the destination object's, the **Replace** action is used. Otherwise, the **Use existing** action is used.
Use older	If the source object's modification time is more recent than the destination object's, the **Use existing** action is used. Otherwise, the **Replace** action is used.
Merge (user/group only)	The privileges, security roles, groups, and Distribution Services addresses and contacts of the source user or group are added to those of the destination user or group.
Do not move (table only)	The selected table is not created in the destination project. This option is only available if the **Allow to override table creation for non-lookup tables that exist only at source project** check box in the **Migration** category of the Object Manager Preferences dialog box is selected.
Force replace (Update packages only)	Replace the object in the destination project with the version of the object in the update package, even if both versions of the object have the same Version ID.
Delete (Update packages only)	Delete the object from the destination project. The version of the object in the update package is not imported into the destination project. **Warning**: If the object in the destination has any used-by dependencies when you import the update package, the import will fail.

Warehouse and other database tables associated with the objects moved are handled in specific ways, depending on your conflict resolution choices. For details, see *Conflict resolution and tables, page 248*.

If you choose to replace a schema object, the following message may appear:

```
The schema has been modified. In order for the changes to
take effect, you must update the schema.
```

This message also appears if you choose to replace an application object that depends on an attribute, and you have made changes to that attribute by modifying its form properties at the report level or its column definition through another attribute. For information about modifying the properties of an attribute, see the Project Design Guide.

To update the project schema, from the Object Manager **Project** menu, select **Update Schema**. For details about updating the project schema, see the *Optimizing and Maintaining your Project* chapter in the Project Design Guide.

To resolve a conflict

1. Select the object or objects that you want to resolve the conflict for. You can select multiple objects by holding down **SHIFT** or **CTRL** when selecting.
2. Choose an option from the **Action** drop-down list (see table above). This option is set for all selected objects.
3. On the toolbar, click **Proceed**. The conflict is resolved according to the selected action.

Setting default actions for conflict resolutions

You can determine the default actions that display in the Conflict Resolution dialog box when a conflict occurs. This includes setting the default actions for the following object categories and types:

- Application objects
- Schema objects
- Configuration objects
- Folders
- Users and user groups

For a list of application, configuration, and schema objects, see *Copying objects, page 234*. For an explanation of each object action, see *Choosing an action to resolve a conflict, page 244*.

You can set a different default action for objects specifically selected by the user, and for objects that are included because they are dependents of selected objects. For example, you can set selected application objects to default to **Use newer** to ensure that you always have the most recent version of any metrics and reports. You can set dependent schema objects to default to **Replace** to use the source project's version of attributes, facts, and hierarchies.

These selections are only the default actions. You can always change the conflict resolution action for a given object when you copy that object.

 © 2016, MicroStrategy Inc.

To set the default conflict resolution actions

1. From the **Tools** menu, select **Object Manager Preferences**. The Object Manager Preferences dialog box opens.
2. Expand the **Conflict Resolution** category, and select **Default Object Actions**.
3. Make any changes to the default actions for each category of objects.
   - For an explanation of the differences between application, configuration, and schema objects, see *Copying objects, page 234*.
   - For an explanation of each object action, see *Choosing an action to resolve a conflict, page 244*.
4. Click **OK**. The Object Manager Preferences dialog box closes and your preferences are saved.

Conflict resolution and access control lists

When you update or add an object in the destination project, by default the object keeps its access control list (ACL) from the source project. You can change this behavior in two ways:

- If you resolve a conflict with the Replace action, and the access control lists (ACL) of the objects are different between the two projects, you can choose whether to keep the existing ACL in the destination project or replace it with the ACL from the source project.
- If you add a new object to the destination project with the Create New or Keep Both action, you can choose to have the object inherit its ACL from the destination folder instead of keeping its own ACL. This is helpful when copying an object into a user's profile folder, so that the user can have full control over the object.

The Use Older or Use Newer actions always keep the ACL of whichever object (source or destination) is used.

To set the ACL options

1. From the **Tools** menu, select **Object Manager Preferences**. The Object Manager Preferences dialog box opens.
2. Expand the **Conflict Resolution** category, and select **Access Control List**.
3. Under **ACL option on replacing objects**, select how to handle the ACL for conflicts resolved with the Replace action:
   - To use the ACL of the source object, select **Keep existing ACL when replacing objects**.

© 2016, MicroStrategy Inc.

- To use the ACL of the replaced destination object, select **Replace existing ACL when replacing objects**.

  If this option is selected, the ACL is replaced even if the source and destination objects are identical.

4. Under **ACL option on new objects**, select how to handle the ACL for new objects added to the destination project:
   - To use the ACL of the source object, select **Keep ACL as in the source objects**.
   - To inherit the ACL from the destination folder, select **Inherit ACL from the destination folder**.
5. Click **OK**. The Object Manager Preferences dialog box closes and your preferences are saved.

Conflict resolution and tables

When an attribute or fact is migrated from one project to another using Object Manager, either specifically or because it is a dependent of another object, by default all dependent tables are also migrated. This includes warehouse tables as well as MDX tables and XDA tables.

You can choose not to create a dependent table in the destination project by changing the Action for the table from **Create New** to **Ignore**. You can also choose not to migrate any dependent tables by specifying that they not be included in Object Manager's dependency search. For detailed information, including instructions, see *Migrating dependent objects, page 240*.

The following list and related tables explain how the attribute - table or fact - table relationship is handled, based on the existing objects and tables and the conflict resolution action you select.

In the following list and tables, attribute, fact, and table descriptions refer to the destination project. For example, "new attribute" means the attribute is new to the destination project: it exists in the source project but not the destination project.

- **New attribute or fact, new table**: There is no conflict resolution. By default the table is moved with the object. You can choose not to create the dependent table in the destination project by changing the Action for the table from **Create New** to **Ignore**.
- **New attribute or fact, existing table**: The object in the source project contains a reference to the table in its definition. The table in the destination project has no reference to the object because the object is not present in the destination project. In this case the new object will have the same references to the table as it did in the source project.
- **Existing attribute or fact, new table**: The object in the destination project does not refer to the table because the table does not exist in the destination project. The object in the source project contains a reference to the table in its definition.

 © 2016, MicroStrategy Inc.

Object Action	What happens in the destination project
Use Existing	The object does not reference the table.
Replace	The object has the same references to the table as it does in the source project.
Keep Both	No change is made to the destination object. The source object is duplicated in the destination project. The duplicated object will have the same references to the table as it does in the source project.

- **Existing attribute or fact, existing table**: The object has a reference to the table in the source project but has no reference to it in the destination project.

Object Action	What happens in the destination project
Use Existing	The object does not reference the table.
Replace	The object has the same references to the table as it does in the source project.
Keep Both	No change is made to the destination object. The source object is duplicated in the destination project. The duplicated object will have the same references to the table as it does in the source project.

- **Existing attribute or fact, existing table**: The object has no reference to the table in the source project but has a reference to it in the destination project.

Object Action	What happens in the destination project
Use Existing	The object has the same references to the table as it did before the action.
Replace	The object does not reference the table.
Keep Both	No change is made to the destination object. The source object is duplicated in the destination project. The duplicated object will not reference the table.

Copying objects in a batch: Update packages

In some cases, you may need to update the objects in several folders at once, or at a time when the source project is offline. Object Manager allows you to save the objects you want to copy in an update package, and import that package into any number of destination projects at a later date.

For example, you have several developers who are each responsible for a subset of the objects in the development project. The developers can submit update packages, with a list of the objects in the packages, to the project administrator. The administrator can then import those packages into the test project to apply the changes from each

developer. If a change causes a problem with the test project, the administrator can undo the package import process.

If your update package includes any schema objects, you may need to update the project schema after importing the package. For more information about updating the schema after importing an update package, see *Update packages and updating the project schema, page 264*.

About update packages

An update package is a file containing a set of object definitions and conflict resolution rules. When you create an update package, you first add objects, and then specify how any conflict involving the objects is resolved. For more information on resolving conflicts with objects, see *Resolving conflicts when copying objects, page 243*.

In addition to the standard Object Manager conflict resolution rules (see *Choosing an action to resolve a conflict, page 244*), two additional rules are available for update packages:

- **Force Replace**: Replace the object in the destination project with the version of the object in the update package, even if both versions of the object have the same Version ID.
- **Delete**: Delete the object from the destination project. The version of the object in the update package is not imported into the destination project.

If the object in the destination has any used-by dependencies when you import the update package, the import will fail.

Object Manager supports the following kinds of update packages:

- **Project update packages** contain application and schema objects from a single project.
- **Configuration update packages** contain configuration objects from a single project source.
    - **Project security update packages** contain security information about users and user groups, such as privileges, security roles, and security filters, for a single project. Since these update packages involve users and groups, which are configuration objects, they are created at the same time as configuration update packages.
- **Undo packages** enable you to reverse the changes made by importing one of the other types of packages. You create undo packages based on existing update packages. For more information about undo packages, including instructions on creating and importing them, see *Rolling back changes: Undo packages, page 265*.

Updating project access information for users and groups

You can include users and groups in a configuration update package. However, the project access information, such as privileges, security roles, and security filters, for those

 © 2016, MicroStrategy Inc.

users and groups is not included in the configuration update package, because this information can be different for each project.

Specifically, configuration update packages do not include the information found in the **Project Access** and **Security Filter** categories of the User Editor or Group Editor. All other user and group information is included in the configuration update package when you add a user or group to the package.

To update your users and groups with the project access information for each project, you must create a project security update package for each project. You create these packages at the same time that you create the configuration update package, by selecting the **Create project security packages** check box and specifying which projects you want to create a project security update package for. For detailed instructions on creating a configuration update package and project security update packages, see *Creating a configuration update package, page 253*.

You must import the configuration update package before importing the project security update packages.

Creating an update package

You create update packages from within Object Manager. From the Create Package dialog box, you select the objects to copy from the source project, and the rules that govern the cases when these objects already exist in the destination project.

You can also create update packages from the command line, using rules specified in an XML file. In the Create Package dialog box, you specify a container object, such as a folder, search object, or object prompt, and specify the conflict resolution rules. Object Manager creates an XML file based on your specifications. You can then use that XML file to create an update package that contains all objects included in the container. For more information and instructions, see *Creating an update package from the command line, page 255*.

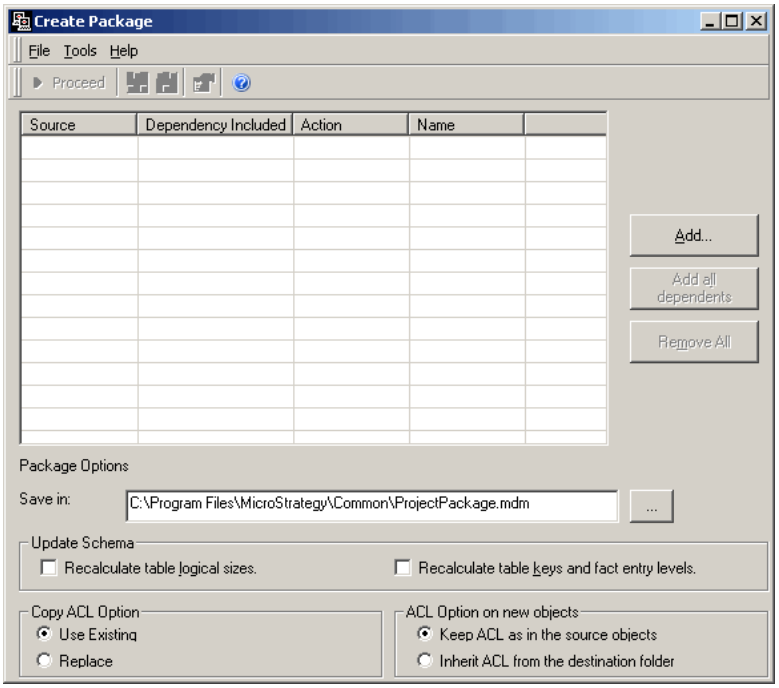
Note the following:

- Configuration update packages and project security update packages are created slightly differently from project update packages. For instructions on how to create a configuration update package and associated project security update packages, see *Creating a configuration update package, page 253*.
- By default, users cannot create project update packages in read-only mode. This is because objects, and their used dependencies, may be changed between the time they are selected for inclusion in the update package and the time the package is actually generated. For more information, see *Project locking with Object Manager, page 233*.

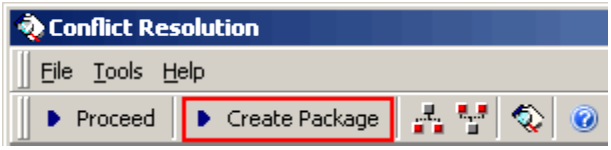
To create a project update package

1 In Object Manager, log in to a project.

2 From the **Tools** menu, select **Create Package**. The Create Package dialog box opens.



You can also open this dialog box from the Conflict Resolution dialog box by clicking **Create Package**. In this case, all objects in the Conflict Resolution dialog box, and all dependents of those objects, are automatically included in the package.



Adding objects to the package

3 To add objects to the package, do one of the following:

- Drag and drop objects from the Object Browser into the Create Package dialog box.
- Click **Add**. An Add Objects dialog box opens. Select the desired objects and click **>**. Then click **OK**.
- Click **Add**. An Add Objects dialog box opens. You can import the results of a previously saved search object.

4 To add the dependents of all objects to the package, click **Add all used dependencies**. All dependent objects of all objects currently listed in the package are added to the package.

 © 2016, MicroStrategy Inc.

If the dependent objects for a specific object do not exist in either the destination project source or in the update package, the update package cannot be applied. If you choose not to add dependent objects to the package, make sure that all dependent objects are included in the destination project source.

5 To add the dependents of specific objects, select those objects, right-click, and select **Add used dependencies**. All dependent objects of those objects are added to the package.

Configuring the package

6 To change the conflict resolution action for an object, double-click the **Action** column for the object and select the new action from the drop-down list. For an explanation of the actions, see *Choosing an action to resolve a conflict, page 244*.

7 Select the schema update options for this package. For more details on these options, see *Update packages and updating the project schema, page 264*.

8 Select the ACL options for objects in this package. For more details on these options, see *Conflict resolution and access control lists, page 247*.

Saving the package

9 Enter the name and location of the package file in the **Save As** field. The default file extension for update packages is `.mmp`.

You can set the default location in the Object Manager Preferences dialog box, in the **Object Manager: Browsing** category.

10 To save a log file containing information about the package's contents in the Object Manager directory, from the **File** menu select **Save As Text File** or **Save As Excel File**.

11 When you have added all objects to the package, click **Proceed**. The package is created in the specified location.

Creating a configuration update package

A configuration update package contains configuration objects from a project source, instead of application and schema objects from a single project. As such, configuration update packages are created at the project source level.

If you choose to include users or groups in a configuration update package, project access information (such as privileges, security roles, and security filters) is not included in the configuration package. To migrate project access information about the users or groups, you must create a project security update package for each project at the same time you create the configuration update package. For more information about project security packages, see *Updating project access information for users and groups, page 250*.

© 2016, MicroStrategy Inc.

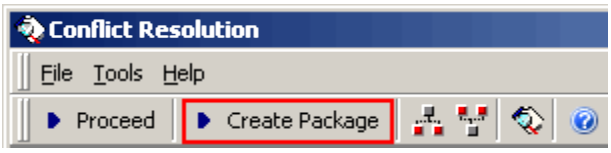
To create a configuration update package

1 In Object Manager, log in to a project source.

2 In the folder list, select the top-level project source.

3 From the **Tools** menu, select **Create Configuration Package**. The Create Package dialog box opens.

You can also open this dialog box from the Conflict Resolution dialog box by clicking **Create Package**. In this case, all objects in the Conflict Resolution dialog box, and all dependents of those objects, are automatically included in the package.



Adding configuration objects to the package

4 To add configuration objects to the package, click **Add Configuration Objects**. A Configuration - Search Objects dialog box opens.

5 Search for the objects you want to add to the package. For instructions on performing a search, see the Help.

6 When the objects are loaded in the search area, click and drag them to the Create Package dialog box.

7 When you have added all the desired objects to the package, close the Configuration - Search for Objects dialog box.

8 To add the dependents of all objects to the package, click **Add all used dependencies**. All dependent objects of all objects currently listed in the package are added to the package.

If the dependent objects for a specific object do not exist in either the destination project source or in the update package, the update package cannot be applied. If you choose not to add dependent objects to the package, make sure that all dependent objects are included in the destination project source.

9 To add the dependents of specific objects, select those objects and click **Add used dependencies**. All dependent objects of those objects are added to the package.

Creating packages for project-level user and group access

10 If your project includes users or groups, and you want to include project-level information about those users or groups, select the **Create project security packages**

 © 2016, MicroStrategy Inc.

check box. For information about project security packages, see *Updating project access information for users and groups, page 250*.

11 In the **Projects** area, select the check boxes next to the projects you want to create project security packages for.

Configuring the package

12 To change the conflict resolution action for an object, double-click the **Action** column for the object and select the new action from the drop-down list. For an explanation of the actions, see *Choosing an action to resolve a conflict, page 244*.

If you are creating project security update packages, you must select **Replace** as the conflict resolution action for all users and groups. Otherwise the project-level security information about those users and groups is not copied into the destination project.

13 Select the ACL options for objects in this package. For more details on these options, see *Conflict resolution and access control lists, page 247*.

Saving the package

14 Enter the name and location of the package file in the **Save As** field. The default file extension for update packages is `.mmp`.

Project security update packages are named *`ProjectSource_ProjectName`*`.mmp`, and are created in the same location as the configuration update package.

15 To save a log file containing information about the package's contents in the Object Manager directory, from the **File** menu select **Save As Text File** or **Save As Excel File**.

16 When you have added all objects to the package, click **Proceed**. The configuration update package and any associated project security update packages are created in the specified location.

Creating an update package from the command line

You may want to schedule the creation of an update package at a later time, so that the project is not locked during normal business hours. Or you may want to create a package containing certain objects on a specific schedule. For example, you may want to create a new package every week that contains all the new metrics from the development project.

You can use Object Manager to create an XML file specifying what objects are to be included in the update package. That XML file can then be used to create the package from the command line.

The XML file specifies a container object in the source project, that is, a folder, search object, or object prompt. When you create the package from the XML file, all objects

included in that container object are included in the update package, as listed in the table below:

If the XML file specifies a...	The update package contains...
Folder	All objects in the folder
Search object	All objects returned by the search
Object prompt	All objects returned by the prompt

To create an XML file for a configuration update package, see *Manually creating an update package creation XML file, page 258*. You cannot create a configuration update package XML file from within Object Manager because container objects do not exist at the project source level.

To create an XML file for creating an update package from the command line

1. In Object Manager, log in to a project.
2. From the **Tools** menu, select **Create Package**. The Create Package dialog box opens.

Adding a container object to the package

3. Click **Add**. The Add Objects dialog box opens.
4. You need to specify what to use as a container object. You can use a search object, object prompt, or folder. To specify a search object or object prompt as the container object:
   a. Make sure the **Import selected objects** option is selected.
   b. In the **Available objects** area, browse to the search object or object prompt.
   c. Select the search object or object prompt and click **>**.
5. OR, to specify a folder as the container object:
   a. Select the **Import folder and children recursively** option.
   b. Type the name of the folder in the field, or click **...** (the browse button) and browse to the folder.
6. Select the **Return as a container to create XML** check box.
7. Click **OK**. The Add Objects dialog box closes.

 © 2016, MicroStrategy Inc.

8 To add the dependents of all objects to the package, select the **Add all used dependencies** check box. All dependent objects of all objects included in the container object will be included in the package when it is created.

If the dependent objects for a specific object do not exist in either the destination project or in the update package, the update package cannot be applied. If you choose not to include dependent objects in the package, make sure that all dependent objects are included in the destination project.

Configuring the package

9 To change the conflict resolution action for an object, double-click the **Action** column for the object and select the new action from the drop-down list. For an explanation of the actions, see *Choosing an action to resolve a conflict, page 244*.

10 Select the schema update options for this package. For more details on these options, see *Update packages and updating the project schema, page 264*.

11 Select the ACL options for objects in this package. For more details on these options, see *Conflict resolution and access control lists, page 247*.

Saving the XML file

12 Enter the name and location of the package file to be created by this XML in the **Save As** field. The default file extension for update packages is `.mmp`.

You can set the default location in the Object Manager Preferences dialog box, in the **Object Manager: Browsing** category.

13 Click **Create XML**. You are prompted to type the name and location of the XML file. By default, this is the same as the name and location of the package file, with an `.xml` extension instead of `.mmp`.

14 Click **Save**. The XML file is saved.

To create an update package from an XML file

Creating a package from the command line locks the project metadata for the duration of the package creation. Other users cannot make any changes to the project until it becomes unlocked. For detailed information about the effects of locking a project, see *Locking projects, page 231*.

Call the Project Merge executable, `projectmerge.exe`, with the following parameters:

© 2016, MicroStrategy Inc.

Effect	Parameter
Use this XML file to create an update package (required)	`-f` *Filename.xml*
Log into the project source with this password (the login ID to be used is stored in the XML file)	`-sp`*Password*
Log into the project with this password (the login ID to be used is stored in the XML file)	`-smp` *Password*
Suppress status updates (useful for creating an update package in the background, so that the status window does not appear)	`-sup`

Manually creating an update package creation XML file

You can also create the XML file to create an update package without opening Object Manager. To do this, you first copy a sample XML file that contains the necessary parameters, and then edit that copy to include a list of the objects to be migrated and conflict resolution rules for those objects.

This is the only way to create an XML file to create a configuration update package.

Sample package creation XML files for project update packages and configuration update packages are in the Object Manager folder. By default this folder is `C:\Program Files (x86)\MicroStrategy\Object Manager\`.

The XML file has the same structure as an XML file created using the Project Merge Wizard. For more information about creating an XML file for use with Project Merge, see *Merging projects to synchronize objects, page 266*.

High-level steps to manually create an update package creation XML file

1. Make a copy of one of the sample XML files:
   - To create a project update package, copy the file `createProjectPackage.xml`.
   - To create a configuration update package, copy the file `createConfigPackage.xml`.
2. Edit your copy of the XML file to include the following information, in the appropriate XML tags:
   - **SearchID** (project update package only): The GUID of a search object that returns the objects to be added to the project update package.

 © 2016, MicroStrategy Inc.

- **TimeStamp** (configuration update package only): A timestamp, of the form MM/DD/YYYY hh:mm:ss (am/pm). All configuration objects modified after that timestamp are included in the update package.
- **PackageFile**: The name and path of the update package. If a package with this name already exists in this path, the creation timestamp is appended to the name of the package created by this file.
- **AddDependents**:
  - **Yes** for the package to include all dependents of all objects in the package.
  - **No** for the package to only include the specified objects.
- **Location**: In a three-tier system, this is the name of the machine that is used to connect to the project source. In a two-tier system, this is the DSN used to connect to the project source.
- **Project** (project update package only): The project containing the objects to include in the update package.
- **ConnectionMode**:
  - **2-tier** for a direct (2-tier) project source connection.
  - **3-tier** for a server (3-tier) project source connection.
- **AuthenticationMode**: The authentication mode used to connect to the project source, either **Standard** or **Windows**.
- **Login**: The user name to connect to the project source. You must provide a password for the user name when you run the XML file from the command line.

3. For a project update package, you can specify conflict resolution rules for individual objects. In an **Operation** block, specify the **ID** (GUID) and **Type** of the object, and the action to be taken. For information about the actions that can be taken in conflict resolution, see *Choosing an action to resolve a conflict, page 244*.
4. Save the XML file.
5. When you are ready to create the update package from the XML file, call the Project Merge executable, `projectmerge.exe`, as described in *To create an update package from an XML file, page 257*.

Editing an update package

You can make changes to an update package after it has been created. You can remove objects from the package, change the conflict resolution rules for objects in the package, and set the schema update and ACL options for the package.

You cannot add objects to an update package once it has been created. Instead, you can create a new package containing those objects.

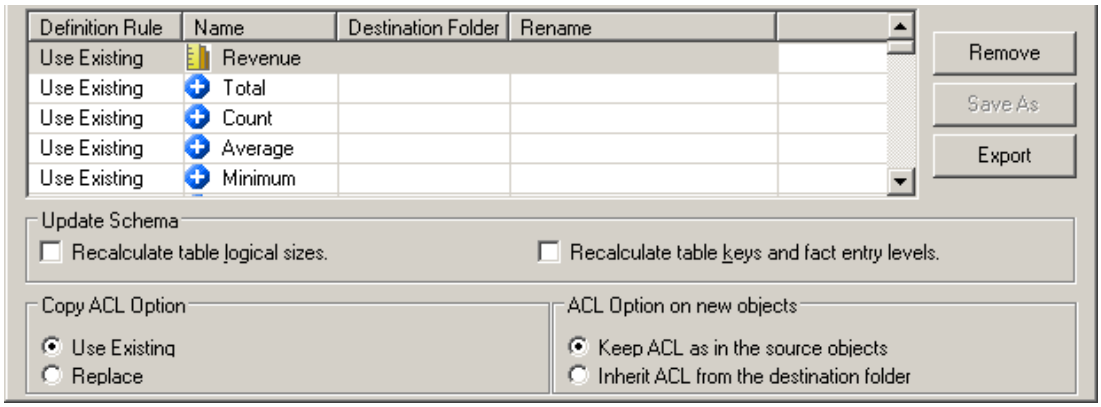
To edit an update package

1 In Object Manager, log in to a project or project source.

2 From the **Tools** menu, select **Import Package** or **Import Configuration Package**. The Import Package or Import Configuration Package dialog box opens.

3 In the **Selected Package** field, type the name and path of the update package, or click **...** (the browse button) to browse to the update package.

4 Click **Edit**. The Editing pane opens at the bottom of the dialog box, as shown below.



5 To change the conflict resolution action for an object, double-click in the **Definition Rule** column for that object and, from the drop-down list, select the new conflict resolution rule.

When you edit a package, the **Create New** action is changed to the **Replace** action.

6 To rename an object in the destination project, double-click in the **Rename** column for that object and type the new name for the object.

7 To remove an object from the update package, select the object and click **Remove**.

8 You can also change the schema update options (for a project update package only) or the access control list conflict resolution options. For information about the schema update options, see *Update packages and updating the project schema, page 264*. For information about the ACL conflict resolution options, see *Conflict resolution and access control lists, page 247*.

9 To create a text file containing a list of the objects in the update package and their conflict resolution actions, click **Export**.

10 When you are done making changes to the update package, click **Save As**. A Save dialog opens for you to save the edited package. The default new name for the update package is the original name of the package with a date and time stamp appended. Click **Save** to save the update package and return to Object Manager.

 © 2016, MicroStrategy Inc.

Importing an update package

An update package is saved in a file, and can be freely copied and moved between machines.

Importing a package causes the project metadata to become locked for the duration of the import. Other users cannot make any changes to the project until it becomes unlocked. For detailed information about the effects of locking a project, see *Locking projects, page 231*.

You can import an update package into a project or project source in the following ways:

- **From within Object Manager**: You can use the Object Manager graphical interface to import an update package.
- **From the command line**: MicroStrategy provides a command line utility for importing update packages. You can use a scheduler such as Windows Scheduler to import the package at a later time, such as when the load on the destination project is light.

  The command line Import Package utility only supports Standard and Windows Authentication. If your project source uses a different form of authentication, you cannot use the Import Package utility to import an update package.

  You can also create an XML file to import an update package from the command line, similar to using an XML file to create an update package as described in *Creating an update package from the command line, page 255*.
- **Using a Command Manager script**: You can also execute a Command Manager script to import an update package without using Object Manager. Command Manager is an administrative tool that enables you to perform various administrative and project development tasks by using text commands that can be saved as scripts. For more information about Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*.

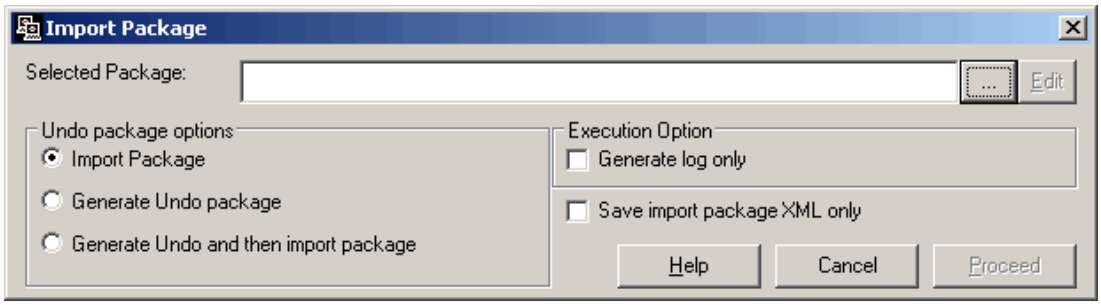
Prerequisites

- If you are importing a package that is stored on a machine other than the Intelligence Server machine, make sure the package can be accessed by the Intelligence Server machine.
- Before importing any project security update packages, you must import the associated configuration update package.

To import an update package from Object Manager

1. In Object Manager, log in to the destination project or project source.
2. From the **Tools** menu, select **Import Package** (for a project update package) or **Import Configuration Package** (for a configuration update package). The Import Package or Import Configuration Package dialog box opens.

© 2016, MicroStrategy Inc.



3 In the **Selected Package** field, type the name and path of the update package, or click ... (the browse button) to browse to the update package.

4 In the **Undo Package Options**, select whether to import this update package, generate an undo package for this update package, or both. For more information about undo packages, see *Rolling back changes: Undo packages, page 265*.

5 To create a log file describing the changes that would be made if the update package were imported, instead of importing the update package, select the **Generate Log Only** check box.

6 Click **Proceed**. All objects in the update package are copied to the destination project or project source, following the rules specified in the update package. A log file containing information about the import process is created in the Object Manager directory.

Any objects that exist in different folders in the update package and the destination project are handled according to the **Synchronize folder locations in source and destination for migrated objects** preference in the **Migration** category in the Object Manager Preferences dialog box.

7 If the package made any changes to the project schema, you may need to update the schema for the changes to take effect. To update the project schema, from the Object Manager **Project** menu, select **Update Schema**.

To import an update package from the command line

Call the Import Package executable, `MAImportPackage.exe`. By default, this file is located in `C:\Program Files (x86)\Common Files\MicroStrategy`. Use the following parameters:

Only Standard Authentication and Windows Authentication are supported by the Import Package utility.

Effect	Parameter
Import package into this project source (required)	`-n`

 © 2016, MicroStrategy Inc.

Effect	Parameter
	ProjectSourceName
Log into the project source with this MicroStrategy username and password, using standard authentication (required unless you are using Windows authentication)	-u *UserName* -p *Password*
Import this package into the specified project source (required) **Note**: The location must be specified relative to the Intelligence Server machine, not relative to the machine running the Import Package utility.	-f *PackageLocation*
Import the package into this project (required for project update packages)	-j *ProjectName*
Log information about the import process to this file **Note**: The location of the log file must be specified relative to the machine running the Import Package utility.	-l *LogLocation*
Force a configuration or project lock prior to importing the package. This lock is released after the package is imported. For more information about project and configuration locking, see *Locking projects, page 231*.	-forcelocking

A full list of parameters can be accessed from a command prompt by entering `importpackage.exe -h`.

To import an update package using an XML file

Create the XML file

1. In Object Manager, log in to the destination project or project source.
2. From the **Tools** menu, select **Import Package** (for a project update package) or **Import Configuration Package** (for a configuration update package). The Import Package or Import Configuration Package dialog box opens.
3. In the **Selected Package** field, type the name and path of the update package, or click ... (the browse button) to browse to the update package.
4. Select the **Save import package XML file** check box.
5. Click **Proceed**. You are prompted to type the name and location of the XML file. By default, this is the same as the name and location of the package file, with an `.xml` extension instead of `.mmp`. Click **Save**. The XML file is saved.

Import the package from the command line

6. When you are ready to import the update package, call the Project Merge executable, `projectmerge.exe`, with the following parameters:

Effect	Parameter
Use this XML file to import an update package (required)	`-f` *Filename.xml*
Log into the project source with this password (the login ID to be used is stored in the XML file)	`-sp`*Password*
Log into the project with this password (the login ID to be used is stored in the XML file)	`-smp` *Password*
Suppress status updates (useful for importing an update package in the background, so that the status window does not appear)	`-sup`

To import an update package using Command Manager

Call a Command Manager script that contains the following command:

```
IMPORT PACKAGE "Filename.mmp" [FOR PROJECT
"ProjectName"];
```

where "*Filename*" is the name and location of the update package, and "*ProjectName*" is the name of the project that the update is to be applied to.

If the package made any changes to the project schema, you need to update the schema for the changes to take effect. The syntax for updating the schema in a Command Manager script is

```
UPDATE SCHEMA [REFRESHSCHEMA] [RECALTABLEKEYS]
[RECALTABLELOGICAL] [RECALOBJECTCACHE] FOR PROJECT
"ProjectName";
```

Update packages and updating the project schema

If a project update package contains new or replacement schema objects, then when the package is imported the user must update the in-memory definitions of these objects. This is done by updating the project schema.

When you create an update package, you can configure it to automatically perform the following schema update functions:

- **Recalculate table keys and fact entry levels**, if you changed the key structure of a table or if you changed the level at which a fact is stored.
- **Recalculate table logical sizes**, to override any modifications that you have made to logical table sizes. (Logical table sizes affect how the MicroStrategy SQL Engine determines which tables to use in a query.)

 © 2016, MicroStrategy Inc.

The update package cannot recalculate the object client cache size, and it cannot update the schema logical information. These tasks must be performed manually. So, for example, if you import an attribute that has a new attribute form, you must manually update the project schema before any objects in the project can use that attribute form.

You can update the project schema in the following ways:

- In Object Manager, select the project and, from the **Project** menu, select **Update Schema**.
- In Developer, log into the project and, from the **Schema** menu, select **Update Schema**.
- Call a Command Manager script with the following command:

```
UPDATE SCHEMA [REFRESHSCHEMA] [RECALTABLEKEYS]
[RECALTABLELOGICAL] [RECALOBJECTCACHE] FOR PROJECT
"projectname";
```

Updating the schema can also be accomplished by unloading and reloading the project. For information on loading and unloading projects, see *Setting the status of a project, page 27*.

For more detailed information about updating the project schema, see the *Optimizing and Maintaining your Project* chapter in the Project Design Guide.

Rolling back changes: Undo packages

You can use undo packages to roll back the changes made by an update package. An undo package is an automatically created update package consisting of all the objects in an update package, as they are currently configured in the destination project. For example, if you create an undo package for an update package containing a new version of three metrics, the undo package contains the version of those three metrics that currently exists in the destination project.

When you import an update package, you have the option of creating an undo package at the same time as the import. Alternately, you can choose to create an undo package without importing the associated update package.

You import an undo package in the same way as you import any update package. When you import an undo package, the Version ID and Modification Date of all objects in the undo package are restored to their values before the original update package was imported.

The Intelligence Server change journal records the importing of both the original update package and the undo package. Importing an undo package does not remove the change journal record of the original update package. For more information about the change journal, see *Monitoring system activity: Change journaling, page 281*.

Merging projects to synchronize objects

You can use MicroStrategy Project Merge to synchronize a large number of objects between projects. Project Merge streamlines the task of migrating objects from one project to another. While you can use Object Manager to copy objects individually, Project Merge can be used as a bulk copy tool. For differences between Object Manager and Project Merge, see *Comparing Project Merge to Object Manager, page 231*.

The rules that you use to resolve conflicts between the two projects in Project Merge can be saved to an XML file and reused. You can then execute Project Merge repeatedly using this rule file. This allows you to schedule a project merge on a recurring basis. For more details about scheduling project merges, see *Scheduling a project merge, page 271*.

Project Merge migrates an entire project. All objects are copied to the destination project. Any objects that are present in the source project but not the destination project are created in the destination project.

Note the following:

- If you want to merge two projects, MicroStrategy recommends that the projects have related schemas. This means that either one project must be a duplicate of the other, or both projects must be duplicates of a third project. For information about duplicating projects, including instructions, see *Duplicating a project, page 225*.
- To merge two projects that do not have related schemas, the projects must either have been created with MicroStrategy 9.0.1 or later, or have been updated to version 9.0.1 or later using the **Perform system object ID unification** option. For information about this upgrade, see the Upgrade Guide.
- Project Merge does not transfer user and group permissions on objects. To migrate permissions from one project to another, use a project security update package. For more information, see *Updating project access information for users and groups, page 250*.

Projects may need to be merged at various points during their life cycle. These points may include:

- Migrating objects through development, testing, and production projects as the objects become ready for use.
- Receiving a new version of a project from a project developer.

In either case, you must move objects from development to testing, and then to the production projects that your users use every day.

Topics covered in this section include:

- *What happens when you merge projects, page 267*
- *Merging projects with the Project Merge Wizard, page 267*
- *Running Project Merge from the command line, page 268*

 © 2016, MicroStrategy Inc.

- *Scheduling a project merge, page 271*
- *Resolving conflicts when merging projects, page 271*

What happens when you merge projects

Project Merge requires a source project, a destination project, and a set of rules to resolve object conflicts between the two projects. This set of rules is defined in the Project Merge Wizard or loaded from an XML file.

In the MicroStrategy system, every object has an ID (or GUID) and a version. (To see the ID and version of an object, right-click the object and select **Properties**.) Project Merge checks the destination project for the existence of every object in the source project, by ID. The resulting possibilities are described below:

- If an object ID does not exist in the destination project, the object is copied from the source project to the destination project.
- If an object exists in the destination project and has the same object ID and version in both projects, the objects are identical and a copy is not performed.
- If an object exists in the destination project and has the same object ID in both projects but a different version, there is a conflict that must be resolved. The conflict is resolved by following the set of rules specified in the Project Merge Wizard and stored in an XML file. The possible conflict resolutions are discussed in *Project Merge conflict resolution rules, page 272*.

Merging projects with the Project Merge Wizard does not update the modification date of the project, as shown in the Project Configuration Editor. This is because, when copying objects between projects, only the objects themselves change. The definition of the project itself is not modified by Project Merge.

Merging projects with the Project Merge Wizard

The Project Merge Wizard allows you to specify rules and settings for a project merge. For details about all settings available when running the wizard, see the Help. For information about the rules for resolving conflicts, see *Resolving conflicts when merging projects, page 271*.

After going through the steps in the wizard, you can either execute the merge right away or save the rules and settings in a Project Merge XML file. You can use this file to run Project Merge from the Windows command prompt (see *Running Project Merge from the command line, page 268*) or to schedule a merge (see *Scheduling a project merge, page 271*).

Before you use Project Merge in a server (three-tier) environment, check the project source time out setting. In Developer, right-click on the project source and select **Modify Project Source** to open the **Project Source Manager**. On the **Connection** tab, either disable the **Connection times out after** setting by clearing its check box, or else enter a sufficient number of minutes for when the connection should time out,

considering how long the merge processes may take based on the size of the projects. If you are unsure about a setting and have noticed other processes taking a long time, it is recommended you disable the time out setting.

The following scenario runs through the Project Merge Wizard several times, each time fine-tuning the rules, and the final time actually performing the merge.

To safely perform a project merge

Both the source and the destination project must be loaded for the project merge to complete. For more information on loading projects, see *Setting the status of a project, page 27*.

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Project Merge Wizard**. The Project Merge Wizard opens.
2. Follow the steps in the wizard to set your options and conflict resolution rules.

   For details about all settings available when running the wizard, see the Help (press **F1** from within the Project Merge Wizard). For information about the rules for resolving conflicts, see *Resolving conflicts when merging projects, page 271*.
3. Near the end of the wizard, when you are prompted to perform the merge or generate a log file only, select **Generate log file only**. Also, choose to **Save Project Merge XML**. At the end of the wizard, click **Finish**. Because you selected to generate a log file only, this serves as a trial merge.
4. After the trial merge is finished, you can read through the log files to see what would have been copied (or not copied) if the merge had actually been performed.
5. Based on what you learn from the log files, you may wish to change some of the conflict resolution rules you set when going through the wizard. To do this, run the wizard again and, at the beginning of the wizard, choose to **Load Project Merge XML** that you created in the previous run. As you proceed through the wizard, you can fine-tune the settings you specified earlier. At the end of the wizard, choose to **Generate the log file only** (thereby performing another trial) and choose **Save Project Merge XML**. Repeat this step as many times as necessary until the log file indicates that objects are copied or skipped as you desire.
6. When you are satisfied that no more rule changes are needed, run the wizard a final time. At the beginning of the wizard, load the Project Merge XML as you did before. At the end of the wizard, when prompted to perform the merge or generate a log file only, select **Perform merge and generate log file**.

Running Project Merge from the command line

A Project Merge can be launched from the Windows command line. You can also run several sessions of the Project Merge Wizard with the same source project, using the

 © 2016, MicroStrategy Inc.

command prompt. For information on running multiple sessions, see *Multiple project merges from the same project, page 270*.

The settings for this routine must be saved in an XML file which can easily be created using the Project Merge Wizard. Once created, the XML file serves as the input parameter to the command.

The syntax for the `projectmerge.exe` command is shown below. The syntax for the command is simplified.

```
projectmerge -f[ ] -sp[ ] -dp[ ] -smp[ ] -dmp[ ] -sup[ ]
-MD -SU -lto -h
```

All command line parameters are described in the table below.

Parameter	Description and use
`-f[ ]`	Specifies the path and file name (without spaces) of the XML file to use. (You must have already created the file using the Project Merge Wizard.) Example: `-fc:\files\merge.xml`
`-sp[ ]`	Password for SOURCE Project Source. (The login ID to be used is stored in the XML file.) Example: `-sphello`
`-dp[ ]`	Password for DESTINATION Project Source. (The login ID to be used is stored in the XML file.) Example: `-dphello`
`-smp[ ]`	Password for SOURCE metadata. (The login ID to be used is stored in the XML file.) Example: `-smphello`
`-dmp[ ]`	Password for DESTINATION metadata. (The login ID to be used is stored in the XML file.) Example: `-dmphello`
`-sup`	Suppress progress window. This is useful for running a project merge in the background, and the window displaying status of the merge does not appear.
`-MD`	Forces metadata update of DESTINATION metadata if it is older than the SOURCE metadata. Project Merge will not execute unless DESTINATION metadata is the same version as or more recent than SOURCE metadata.
`-SU`	Updates the schema of the DESTINATION project after the Project Merge is completed. This update is required when you make any changes to schema objects (facts, attributes, or hierarchies). **Note**: Do not use this switch if the Project Merge configuration XML contains an instruction to update the schema.
`-lto`	Take ownership of any metadata locks that exist on the source or destination projects. For more information about metadata locking, see *Locking projects, page 231*.
`-h`	Displays help and explanations for all of the above parameters.

A sample command using this syntax is provided below. The command assumes that "`hello`" is the password for all the project source and database connections. The login IDs used with these passwords are stored in the XML file created by the Project Merge Wizard.

```
projectmerge -fc:\temp\merge.xml -sphello -dphello -
smphello -dmphello -lto -MD -SU
```

If the XML file contains a space in the name or the path, you must enclose the name in double quotes, such as:

```
projectmerge -f "c:program files (x86)\xml\
merge.xml" -sphello -dphello -smphello -dmphello -MD
-SU
```

Multiple project merges from the same project

The Project Merge Wizard can perform multiple simultaneous merges from the same project source. This can be useful when you wish to propagate a change to several projects simultaneously.

During a multiple merge, the Project Merge Wizard is prevented from locking the projects. This is so that multiple sessions of the wizard can access the source projects. You will need to manually lock the source project before beginning the merge. You will also need to manually lock the destination projects at the configuration level before beginning the merge. Failing to do this may result in errors in project creation due to objects being changed in the middle of a merge. For information on locking and unlocking projects, see *Locking projects, page 231*. For detailed instructions on how to manually lock and unlock projects, see the *MicroStrategy Developer Help*.

To do this, you must modify the Project Merge XML file, and then make a copy of it for each session that you want to run.

To execute multiple simultaneous merges from one project

1. In a text editor, open the Project Merge Wizard XML file.
2. In the `OMOnOffSettings` section of the file, add the following node: `<Option><ID>OMOnOffSettings</ID><SkipProjectMergeSourceLockingSkipProjectMergeDestConfigLocking/></Option>`.
3. Make one copy of the XML file for each session of the Project Merge Wizard you wish to run.
4. In each XML file, make the following changes:
   - Correct the name of the destination project.
   - Ensure that each file uses a different Project Merge log file name.
5. Manually lock the source project. For detailed steps on locking projects manually, see the *MicroStrategy Developer Help*.
6. Manually lock the destination projects at the configuration level. For detailed steps on locking projects manually, see the *MicroStrategy Developer Help*.

 © 2016, MicroStrategy Inc.

7 For each XML file, run one instance of the Project Merge Wizard from the command line.

Scheduling a project merge

To schedule a delayed or recurring Project Merge, use the `AT` command, which is part of the Microsoft Windows operating system. For instructions on how to use the AT command, refer to the Microsoft Windows help. The sample AT command below schedules Project Merge to run at 6:00 PM (`18:00`) every Friday (`/every:F`).

```
at 18:00 /every:F projectmerge -fc:\temp\merge.xml -
sphello -dphello -smphello -dmphello -MD -SU
```

For a list of the syntax options for this command, see *Running Project Merge from the command line, page 268*.

To schedule a project merge using the Windows command prompt

1 From the Microsoft Windows machine where Project Merge is installed, from the **Start** menu, select **Programs**, then choose **Command Prompt**.

2 Change the drive to the one on which the Project Merge utility is installed. The default installation location is the C: drive (the prompt appears as: `C:\>`)

3 Type an `AT` command that calls the `projectmerge` command. For a list of the syntax options for this command, see *Running Project Merge from the command line, page 268*.

Resolving conflicts when merging projects

Conflicts occur when a destination object's version differs from the source object's version. This difference usually means that the object has been modified in one or both of the projects. These conflicts are resolved by following a set of rules you define as you step through the Project Merge Wizard.

When you define the rules for Project Merge to use, you first set the default conflict resolution action for each category of objects (schema, application, and configuration). (For a list of objects included in each category, see *Copying objects*.) Then you can specify conflict resolution rules at the object type level (attributes, facts, reports, consolidations, events, schedules, and so on). Object type rules override object category rules. Next you can specify rules for specific folders and their contents, which override the object type and object category rules. Finally you can specify rules for specific objects, which, in turn, override object type rules, object category rules, and folder rules.

For example, the **Use Newer** action replaces the destination object with the source object if the source object has been modified more recently than the destination object. If you specified the **Use newer** action for all metrics, but the Sales metric has been changed recently and is not yet ready for the production system, you can specify **Use existing** (use the object in the destination project) for that metric only and it will not be replaced.

© 2016, MicroStrategy Inc.

Project Merge conflict resolution rules

If the source object has a different version than the destination object, that is, the objects exist differently, you must determine what action should occur. The various actions that can be taken to resolve conflicts are explained in the table below.

Action	Effect
Use existing	No change is made to the destination object. The source object is not copied.
Replace	The destination object is replaced with the source object. **Note**: Non-empty folders in the destination location will never have the same version ID and modification time as the source, because the folder is copied first and the objects are added to it, thus changing the version ID and modification times during the copy process.
Keep both	No change is made to the destination object. The source object is duplicated in the destination location.
Use newer	If the source object's modification time is more recent than the destination object's, the **Replace** action is used. Otherwise, the **Use existing** action is used.
Use older	If the source object's modification time is more recent than the destination object's, the **Use existing** action is used. Otherwise, the **Replace** action is used.

Comparing and tracking projects

Often during the project life cycle, you do not know exactly which objects need to be moved from one project to another. This is because there are many developers working on a project and it is difficult for a single person to know all of the work that has been done. The migration process becomes much easier if you first compare objects in the source and destination projects.

You can use the MicroStrategy Project Comparison Wizard to compare objects in related projects. This wizard tells you which objects are different between the two projects, and which objects exist in one project but not in the other. From this list you can decide what objects to move between projects, using Object Manager. For instructions on moving objects with Object Manager, see *Copying objects between projects: Object Manager, page 232*.

You can track changes to your projects with the MicroStrategy Search feature, or retrieve a list of all unused objects in a project with the Find Unreferenced Objects feature of Object Manager.

This section covers the following topics:

- *Comparing objects between two projects, page 273*
- *Tracking your projects with the Search Export feature, page 274*
- *Listing unused objects in a project, page 275*

 © 2016, MicroStrategy Inc.

Comparing objects between two projects

The Project Comparison Wizard compares objects in a source project and a destination project.

For the source project, you specify whether to compare objects from the entire project, or just from a single folder and all its subfolders. You also specify what types of objects (such as reports, attributes, or metrics) to include in the comparison.

Every object in a MicroStrategy project has a unique ID. Project Comparison looks at each object ID in the source project, and compares it to the object in the destination project with the same ID. For each object ID, Project Comparison indicates whether the object is:

- Identical in both projects
- Identical in both projects except for the folder path
- Only present in the source or destination project
- Different between projects, and newer in the source or destination project

You can print this result list, or save it as a text file or an Excel file.

Since the Project Comparison Wizard is a part of Object Manager, you can also select objects from the result set to immediately migrate from the source project to the destination project. For more information about migrating objects using Object Manager, see *Copying objects between projects: Object Manager, page 232*.

Using the Project Comparison Wizard

The following high-level procedure provides an overview of what the Project Comparison Wizard does. For an explanation of the information required at any given page in the wizard, see the Help (from the wizard, click **Help**, or press **F1**).

To compare two projects

Note the following:

- To compare two projects with the Project Comparison Wizard, those projects must have related schemas. This means that either one project must be a duplicate of the other, or both projects must be duplicates of a third project. For information about duplicating projects, including instructions, see *Duplicating a project, page 225*.
- The Project Comparison Wizard is a part of Object Manager, and thus requires the Use Object Manager privilege to run. For an overview of Object Manager, see *Copying objects between projects: Object Manager, page 232*.

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **Object Manager**. Object Manager opens.

© 2016, MicroStrategy Inc.

2. Open a project source in Object Manager.
3. From the **Project** menu, select **Compare Projects**. The Project Comparison Wizard opens.
4. Select the source and destination projects.
5. Specify whether to compare all objects or just objects in a specific folder, and what types of objects to compare.
6. Review your choices at the summary screen and click **Finish**. The objects in the two projects are compared and the Project Comparison Result Set dialog opens. This dialog lists all the objects you selected and the results of their comparison.
7. To save the results, from the **File** menu select **Save as Text File** or **Save as Excel File**.
8. To migrate objects from the source project to the destination project using Object Manager, select those objects in the list and click **Proceed**. For more information about Object Manager, see *Copying objects between projects: Object Manager, page 232*.

Tracking your projects with the Search Export feature

Exporting the results of a search object can be a useful way to keep track of changes to a project. The Search Export feature enables you to perform a search for either a specific object in a project or for a group of objects that meet certain criteria. After the search is performed, you can save your search definition and search results to a text file, and save the search object itself for later reuse.

For example, you can create a search object in the development project that returns all objects that have been changed after a certain date. This lets you know what objects have been updated and need to be migrated to the test project. For more information about development and test projects, see *The project life cycle, page 220*.

The search export file contains the following information:

- The user who was logged in when the search was performed.
- The search type, date and time, and project name.
- Any search criteria entered into the tabs of the Search for Objects dialog box.
- Any miscellaneous settings in Developer that affected the search (such as whether hidden and managed objects were included in the search).
- A list of all the objects returned by the search, including any folders. The list includes object names and paths (object locations in the Developer interface).

 © 2016, MicroStrategy Inc.

To search for objects and save the results in a text file

1. In Developer, from the **Tools** menu, select **Search for Objects**. The Search for Objects dialog box opens.
2. Perform your search. For information on how to configure a search in Developer, see the *MicroStrategy Developer Help*.
3. After your search is complete, from the **Tools** menu in the Search for Objects dialog box, select **Export to Text**. The text file is saved by default to `C:\Program Files (x86)\MicroStrategy\Desktop\ SearchResults_<date and timestamp>.txt`, where *<date and timestamp>* is the day and time when the search was saved. For example, the text file named `SearchResult_022607152554.txt` was saved on February 26, 2007, at 15:25:54, or 3:25 PM.

Listing unused objects in a project

In Object Manager, you can retrieve a list of all the objects in a project that are not used by any other objects. For example, you can find which attributes or metrics are no longer used in any reports, so that you can delete those objects.

To find unused objects

Finding unused objects is a part of Object Manager, and thus requires the Use Object Manager privilege to run. For an overview of Object Manager, see *Copying objects between projects: Object Manager, page 232*.

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **Object Manager**. Object Manager opens.
2. Open a project source in Object Manager.
3. From the **Tools** menu, select **Find Unreferenced Objects**. The Search for Objects dialog box opens.
4. In the **Look In** field, enter the folder you want to start your search in.
5. Make sure the **Include Subfolders** check box is selected.
6. Click **Find Now**. The unused objects are listed at the bottom of the dialog box.

© 2016, MicroStrategy Inc.

Deleting unused schema objects: managed objects

MicroStrategy projects contain schemas and related schema objects, including attributes, tables, hierarchies, and so on. For an introduction to schema objects, see the Project Design Guide.

Certain MicroStrategy features automatically create new schema objects, referred to as managed objects, which are not directly related to the project schema. The features that create their own managed objects are:

- Freeform SQL and Query Builder. For information on Freeform SQL and Query Builder, see the Advanced Reporting Guide.
- MDX cube sources such as SAP BW, Hyperion Essbase, Microsoft Analysis Services, and IBM Cognos TM1. For information on MDX cube sources, see the MDX Cube Reporting Guide.
- Import Data, which lets you use MicroStrategy Web to import data from different data sources, such as an Excel file, a table in a database, or the results of a SQL query, with minimum project design requirements. For more information on how to use the Import Data feature, refer to the MicroStrategy Web Help.

Managed objects are stored in a special system folder, and can be difficult to delete individually due to how these objects are created and stored. If you use one of the features listed above, and then decide to remove some or all of that feature's related reports and MDX cubes from the project, there may be unused managed objects included in your project that can be deleted.

This section covers the following topics:

- *Deleting managed objects one-by-one, page 276*
- *Deleting all unused managed objects, page 277*

Deleting managed objects one-by-one

When you delete managed objects one-by-one, you individually select which managed objects you want to delete and which you want to keep. You can perform this clean-up for any of the Freeform SQL, Query Builder, or MDX cube source database instances included for your project.

For example, you decide to delete a single Freeform SQL report that automatically created a new managed object named Store. When you delete the report, the managed object Store is not automatically deleted. You do not plan to use the object again; however, you do plan to create more Freeform SQL reports and want to keep the database instance included in the project. Instead of deleting the entire Freeform SQL schema, you can delete only the managed object Store.

 © 2016, MicroStrategy Inc.

To delete managed objects one-by-one

1. In Developer, delete any Freeform SQL, Query Builder, or MDX cube reports in the project that depend on the managed objects you want to delete.

   If you are removing MDX cube managed objects, you must also remove any MDX cubes that these managed objects depend on.

2. Right-click the project and select **Search for Objects**. The Search for Objects dialog box opens.
3. From the **Tools** menu, select **Options**. The Search Options dialog box opens.
4. Select the **Display managed objects** and **Display managed objects only** check boxes.
5. Click **OK** to return to the Search for Objects dialog box.
6. Enter your search criteria and select **Find Now**. A list of managed objects appears.
7. Manually delete managed objects by right-clicking their name in the search result and selecting **Delete**.

Deleting all unused managed objects

Managed objects can become unused in a project when you stop using the feature that created the managed objects. You can delete all unused managed objects to clean up your project.

For example, you can create a separate database instance for your Freeform SQL reports in your project. Later on, you may decide to no longer use Freeform SQL, or any of the reports created with the Freeform SQL feature. After you delete all the Freeform SQL reports, you can remove the Freeform SQL database instance from the project. Once you remove the database instance from the project, any Freeform SQL managed objects that depended solely on that database instance can be deleted.

You can implement the same process when removing database instances for Query Builder, SAP BW, Essbase, and Analysis Services.

To delete all unused managed objects from a project

1. Remove all reports created with Freeform SQL, Query Builder, or MDX cubes.

   If you are removing MDX cube managed objects, you must also remove all imported MDX cubes.

2. In Developer, right-click the project and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand the **Database instances** category.

4 Select either **SQL data warehouses** or **MDX data warehouses**, depending on the database instance you want to remove.

Freeform SQL and Query Builder use relational database instances, while SAP BW, Essbase, and Analysis Services use MDX cube database instances. For more information on the difference between the two, see the Installation and Configuration Guide.

5 Clear the check box for the database instance you want to remove from the project. You can only remove a database instance from a project if the database instance has no dependent objects in the project.

6 Click **OK** to accept the changes and close the Project Configuration Editor.

This procedure removes some preliminary object dependencies. Attribute and metric managed objects are not automatically deleted by this procedure, because you can reuse the managed attributes and metrics at a later time. If you do not plan to use the attribute and metric managed objects and want to delete them permanently from your project, continue through the rest of this procedure.

To delete unused attribute and metric managed objects

7 In Developer, from the **Administration** menu, select **Projects**, and then select **Delete unused managed objects**.

 © 2016, MicroStrategy Inc.

7

MONITORING SYSTEM USAGE

Introduction

MicroStrategy provides several ways to track user activity, resource usage, and performance in your system. These include the following:

- *MicroStrategy system monitors, page 279*
- *Monitoring system activity: Change journaling, page 281*
- *Monitoring system usage: Intelligence Server statistics, page 288*
- *Additional monitoring tools, page 299*

MicroStrategy system monitors

You can monitor various aspects of your MicroStrategy system from within Developer. The **Administration** category for a project source contains several system monitors for that project source. These monitors are listed in the table below, and are described in detail in the relevant section of this guide.

For information about monitoring...	See...
Projects loaded on Intelligence Server, or on all nodes of the cluster	*Managing and monitoring projects, page 24*

© 2016, MicroStrategy Inc.

For information about monitoring...	See...
Projects loaded on specific nodes of the cluster	*Managing your projects across nodes of a cluster, page 390*
Jobs that are currently executing	*Monitoring currently executing jobs, page 47*
Users that are currently connected to Intelligence Server	*Monitoring users' connections to projects, page 57*
Active and cached database connections	*Monitoring database instance connections, page 9*
Report and document caches	*Monitoring result caches, page 409*
History List messages	*Monitoring History List messages, page 439*
Intelligent Cubes, whether they are loaded on Intelligence Server	*Managing Intelligent Cubes: Intelligent Cube Monitor, page 458*

Prerequisites

- Before you can view a system monitor, you must have the appropriate privilege to access that monitor. For example, to view the Job Monitor, you must have the Monitor Jobs privilege. For more information about privileges, see *Controlling access to functionality: Privileges, page 66*.
- In addition, you must have Monitoring permission for the server definition that contains that monitor. You can view and modify the ACL for the server definition by right-clicking the **Administration** icon, selecting **Properties**, and then selecting the **Security** tab. For more information about permissions and ACLs, see *Controlling access to objects: Permissions, page 58*.

To view a system monitor

1. In Developer, log in to the project source that you want to monitor. You must log in as a user with the appropriate administrative privilege.
2. Expand the **Administration** category.
3. To monitor projects or clusters, expand the **System Administration** category and select either **Project** or **Cluster Nodes**.
4. To view additional system monitors, expand the **System Monitors** category and select the desired monitor. For a list of the different monitors available, and where you can find more information about each monitor, see the table above.

© 2016, MicroStrategy Inc.

Monitoring system activity: Change journaling

Change journaling is the process of logging information about changes to objects in a project. Change journaling tracks the changes to each object in the system. This makes it easier for administrators to quickly determine when and by whom certain changes were made. For example, reports using a certain metric executed correctly in a test two weeks ago, but no longer execute correctly in this morning's test. The administrator can search the change journal to determine who has made changes to that metric within the last two weeks.

The logged information includes items such as the user who made the change, the date and time of the change, and the type of change (such as saving, copying, or deleting an object). With change journaling, you can keep track of all object changes, from simple user actions such as saving or moving objects to project-wide changes such as project duplication or project merging.

Certain business regulations, such as Sarbanes-Oxley in the United States, require detailed records of changes made to a BI system. Enabling change journaling on all projects in your production environment can aid in compliance with these regulations.

Enabling change journaling

When change journaling is enabled for a project or project source, Intelligence Server logs information in the change journal about any change made to any object in the project or project source. This includes changes made in Developer or MicroStrategy Web as well as through other MicroStrategy tools such as Command Manager or Project Merge.

You can enable change journaling for any number of projects in a project source. For each project, when change journaling is enabled, all changes to all objects in that project are logged.

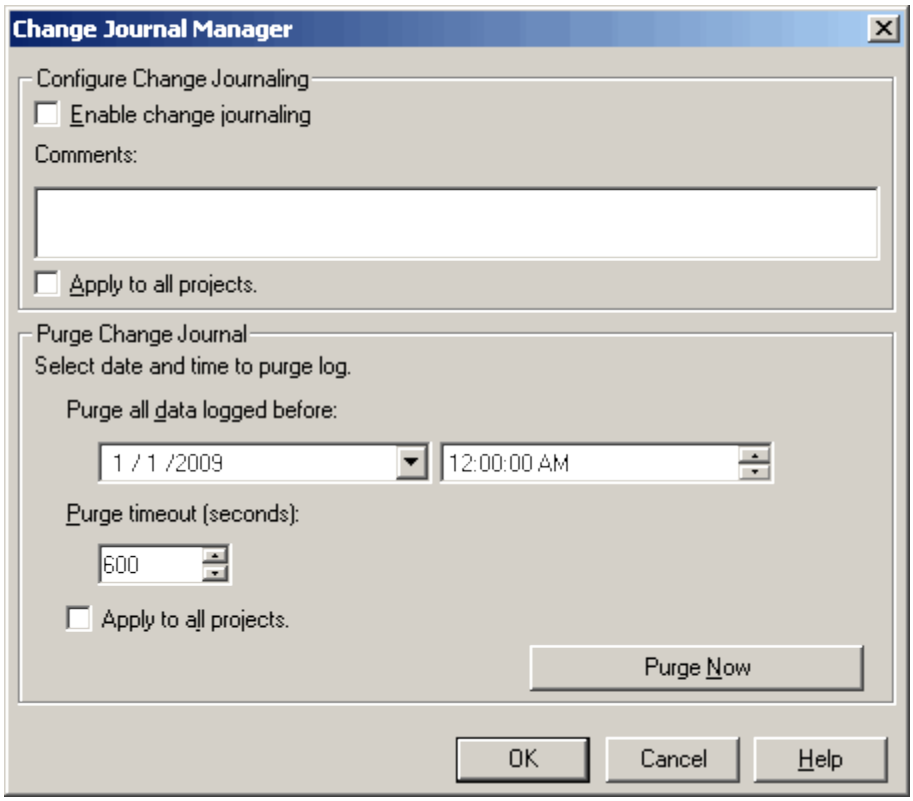
You can also enable change journaling at the project source level. In this case information about all changes to the project configuration objects, such as users or schedules, is logged in the change journal.

By default, change journaling is enabled in all newly created projects and project sources.

To enable or disable change journaling for a project source

1. In Developer, log in to a project source. You must log in as a user with the Configure Change Journaling privilege.
2. Expand **Administration**, and then expand **System Monitors**.
3. Right-click **Change Journal Transactions** and select **Manage Change Journal**. The Change Journal Manager opens.

© 2016, MicroStrategy Inc.



4 To enable or disable change journaling for this project source, select or clear the **Enable change journaling** check box.

5 In the **Comments** field, enter any comments that you may have about the reason for enabling or disabling change journaling.

6 To enable or disable change journaling for all projects in the project source, select the **Apply to all projects** check box. To determine which projects have change journaling on a project-by-project basis, leave this check box cleared.

7 Click **OK**. The Change Journal Manager closes. A transaction is logged in the change journal for each project that enables or disables change journaling.

To enable or disable change journaling for a project

1 From Developer, right-click the project and select **Project Configuration**. The Project Configuration Editor opens.

2 Expand **Project definition**, and then select **Change Journaling**.

3 To enable or disable change journaling for this project, select or clear the **Enable Change Journaling** check box.

4 Click **OK**. The Project Configuration Editor closes. A transaction is logged in the change journal when this project enables or disables change journaling.

 © 2016, MicroStrategy Inc.

Change journal comments

When change journaling is enabled, users are prompted for comments every time they change an object. These comments can provide documentation as to the nature of the changes made to objects.

You can disable the requests for object comments from the Developer Preferences dialog box.

To disable the requests for change journaling comments

1. Open Developer.
2. From the **Tools** menu, select **MicroStrategy Developer Preferences**. The Developer Preferences dialog box opens.
3. Expand **Optional Actions**, and then select **General**.
4. Clear the **Display change journal comments input dialog** check box.
5. Click **OK**. The Developer Preferences dialog box closes. You are no longer prompted to enter a comment when you save objects.

Viewing the change journal entries

When an object is changed, information about the change is entered in the change journal. To view the change journal for all projects in a project source, in Developer, expand **Administration**, then expand **System Monitors**, and then select **Change Journal Transactions**. The change journal entries are listed in the main window of Developer.

You must have the Audit Change Journal privilege to view the change journal.

To view the detailed information for a change journal entry, double-click that entry. Each entry contains the following information:

Entry	Details
Object name	The name of the object that is changed.
Object type	The type of object changed. For example, Metric, User, or Server Definition.
User name	The name of the MicroStrategy user that made the change.
Transaction timestamp	The date and time of the change, based on the time on the Intelligence Server machine.
Transaction type	The type of change and the target of the change. For example, Delete Objects, Save Objects, or Enable Logging.

© 2016, MicroStrategy Inc.

Entry	Details
Transaction source	The application that made the change. For example, Developer, Command Manager, MicroStrategy Web, or Scheduler.
Project name	The name of the project that contains the object that was changed. **Note**: If the object is a configuration object, the project name is listed as <Configuration>
Comments	Any comments entered in the Comments dialog box at the time of the change.
Object ID	The object's GUID, a unique MicroStrategy system identifier.
Machine name	The name of the machine that the object was changed on.
Change type	The type of change that was made. For example, Create, Change, or Delete.
Transaction ID	A unique 32-digit hexadecimal number that identifies this change.
Session ID	A unique 32-digit hexadecimal number that identifies the user session in which the change was made.
Link ID	For MicroStrategy use.

This information can also be viewed in the columns of the change journal. To change the visible columns, right-click anywhere in the change journal and select **View Options**. In the View Options dialog box, select the columns you want to see.

Increasing the number of change journal entries to view or export

By default the change journal displays and exports the last 1,000 entries. You can increase this number in the **Browsing** category of the Developer Preferences dialog box. Viewing more entries may make the browsing and exporting process take longer.

To increase the number of displayed or exported change journal entries

1. In Developer, from the **Tools** menu select **MicroStrategy Developer Preferences**. The Developer Preferences dialog box opens.
2. In the **General** category, select **Browsing**.
3. In the **Maximum number of monitoring objects displayed per page** field, specify the maximum number of change journal entries to display.
4. In the **Maximum number of transactions retrieved per metadata change journaling search** field, specify the maximum number of change journal entries to export.
5. Click **OK**. The dialog box closes and your changes are saved.

 © 2016, MicroStrategy Inc.

Searching the change journal for relevant entries

Because the change journal records every transaction, finding the relevant records can be daunting. To make searching the change journal easier, you can filter it so that you see the relevant entries.

For example:

- To find out when certain users were given certain permissions, you can view entries related to Users.
- To discover which user made a change that caused a report to stop executing correctly, you can view the entries related to that report.

You can also quickly filter the entries so that you see the entries for an object or the changes made by a specific user. To do this, right-click one of the entries for that object or that user and select either **Filter view by object** or **Filter view by user**. To remove the filter, right-click in the change journal and select **Clear filter view**.

To filter the change journal for relevant entries

1. In the Change Journal Transactions Monitor, right-click and select **Filter**. The Change Journal Monitor Filter dialog box opens.
2. To filter the change journal by changed object type, project, transaction type, or source of the change, select from the appropriate drop-down list.
3. To filter the change journal by multiple conditions, click **Advanced**. The advanced filtering options panel opens at the bottom of the dialog box. Enter the columns and conditions.
4. To see changes made in a specific time range, enter the start and end time and date.
5. To view all transactions, not just those that change the version of an object, clear the **Show version changes only** and **Hide Empty Transactions** check boxes.

   If the **Show version changes only** check box is cleared, two transactions named "LinkItem" are listed for every time an application object is saved. These transactions are monitored for MicroStrategy technical support use and do not indicate that the application object has been changed. Any time the object has actually been changed, a SaveObjects transaction with the name of the application object is listed.
6. Click **OK** to close the dialog box and filter the change journal.

To quickly filter the change journal by object or user

1. In the Change Journal Transactions Monitor, right-click an entry for the object or user you want to filter by, and select the type of filtering:

- To see the changes to this object, select **Filter view by object**.
- To see the changes made by this user, select **Filter view by user**.

2 To remove a quick filter, right-click in the change journal and select **Clear filter view**.

Exporting the change journal

You can export the contents of the change journal to a text file. This can be useful so that you can save this file to an archival location, or email it to MicroStrategy technical support for assistance with a problem.

The name of this file is `AuditLog_`*`MMDDYYhhmmss`*`.txt`, where *`MMDDYY`* is the month, date, and last two digits of the year, and *`hhmmss`* is the timestamp, in 24-hour format. This file is saved in the MicroStrategy Common Files directory. By default this directory is `C:\Program Files (x86)\Common Files\MicroStrategy\`.

When you export the change journal, any filters that you have used to view the results of the change journal are also applied to the export. If you want to export the entire change journal, make sure that no filters are currently in use. To do this, right-click in the change journal and select **Clear filter view**.

To export the change journal to a file

1 In Developer, expand **Administration**, and then expand **System Monitors**.

2 Right-click **Change Audit** and select **Export list**. The change journal is exported to a text file.

A prompt is displayed informing you that the list was exported and noting the folder and file name, and asks if you want to view the file. To view the file, click **Yes**.

Purging the change journal

You can keep the size of the change journal to a manageable size by periodically purging older entries that you no longer need to keep.

When you purge the change journal, you specify a date and time. All entries in the change journal that were recorded prior to that date and time are deleted. You can purge the change journal for an individual project, or for all projects in a project source.

MicroStrategy recommends archiving your change journal entries before purging. For instructions on how to archive the change journal, see *Exporting the change journal, page 286*.

 © 2016, MicroStrategy Inc.

To purge the change journal for all projects in a project source

1. In Developer, expand **Administration**, and then expand **System Monitors**.
2. Right-click **Change Journal Transactions** and select **Manage change journal**. The Change Journal Manager opens.
3. Set the date and time. All data recorded before this date and time is deleted from the change journal.
4. To purge data for all projects, select the **Apply to all projects** check box. To purge data relating to the project source configuration, leave this check box cleared.
5. Click **Purge Now**. When the warning dialog box opens, click **Yes** to purge the data, or **No** to cancel the purge. If you click **Yes**, change journal information recorded before the specified date is deleted.

   If you are logging transactions for this project source, a Purge Log transaction is logged when you purge the change journal.
6. Click **Cancel**. The Change Journal Manager closes.

To purge the change journal for a single project

1. In Developer, right-click on the project and select **Project Configuration**. The Project Configuration Editor for that project opens.
2. Expand **Project definition**, and then select **Change Journaling**.
3. Under **Purge Change Journal**, set the date and time. All change journal data for this project from before this date and time will be deleted from the change journal.
4. In the **Purge timeout (seconds)** field, specify the timeout setting in seconds.
5. Click **Purge Now**. When the warning dialog box opens, click **Yes** to purge the data, or **No** to cancel the purge. If you click **Yes**, change journal information for this project from before the specified date and time is deleted.
6. Click **OK**. The Project Configuration Editor closes.

© 2016, MicroStrategy Inc.

Monitoring system usage: Intelligence Server statistics

To tune your system for best performance, you need information about how the system is being used. Intelligence Server can record usage and performance statistics for each project in your system. You can then analyze these statistics to determine what changes need to be made.

This section provides the following information about Intelligence Server statistics:

- *Overview of Intelligence Server statistics, page 288*
- *Best practices for recording Intelligence Server statistics, page 292*
- *Configuring Intelligence Server to log statistics, page 293*

MicroStrategy Enterprise Manager can help you analyze the Intelligence Server statistics data. Enterprise Manager consists of a MicroStrategy project containing a wide variety of reports and dashboards that present the statistics data in an easy-to-understand format. For more information about Enterprise Manager, see the Enterprise Manager chapter in the *MicroStrategy Operations Manager Guide*.

Overview of Intelligence Server statistics

Intelligence Server can record a wide variety of statistics relating to user activity, data warehouse activity, report SQL, and system performance. These statistics are logged in the statistics database (see *The statistics database, page 290*).

The statistics that are logged for each project are set in the Project Configuration Editor, in the **Statistics**: **General** subcategory. The options are as follows:

Statistics logging option	Statistics logged
All basic statistics	User session and project session analysis. This option must be selected for any statistics to be logged.
Report job steps	Detailed statistics on the processing of each report.
Document job steps	Detailed statistics on the processing of each document.
Report job SQL	The generated SQL for all report jobs. **Warning**: This option can create a very large statistics table. Select this option when you need the job SQL data.
Report job tables/columns accessed	Data warehouse tables and columns accessed by each report.

 © 2016, MicroStrategy Inc.

Statistics logging option	Statistics logged
Mobile Clients	Detailed statistics on reports and documents that are executed on a mobile device.
Mobile Clients Manipulations This option is available if **Mobile Clients** is selected	Detailed statistics on actions performed by end users on a mobile client.
Only purge statistics logged from the current Intelligence Server.	Purge statistics from the database if they are from the Intelligence Server you are now using. This is applicable if you are using clustered Intelligence Servers.

You can log different statistics for each project. For example, you may want to log the report job SQL for your test project when tracking down an error. If you logged report job SQL for your production project, and your users are running many reports, the statistics database would quickly grow to an unwieldy size.

Recording performance counters in the statistics tables

Intelligence Server can be configured to collect performance information from the Diagnostics and Performance Logging Tool and record that information in the statistics database. For more information about logging performance counters, see *Performance configuration, page 736*.

Intelligence Server can collect and log information from the MicroStrategy Server Jobs and MicroStrategy Server Users categories. On UNIX or Linux, Intelligence Server can also collect and log information from the following categories:

- Memory
- System
- Process
- Processor
- Network Interface
- Physical Disk

This information is recorded in the STG_IS_PERF_MON_STATS table in the statistics database.

To configure the performance counters to record information in the statistics repository

1. Open the Diagnostics and Performance Logging Tool.
   - From Developer: From the **Tools** menu, select **Diagnostics**.

If the **Diagnostics** option does not appear on the **Tools** menu, it has not been enabled. To enable this option, from the **Tools** menu, select **MicroStrategy Developer Preferences**. In the **General** category, in the **Advanced** subcategory, select the **Show Diagnostics Menu Option** check box and click **OK**.

- In Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Diagnostics Configuration**.
- In UNIX/Linux: Navigate to the directory `~/MicroStrategy/bin` and enter `mstrdiag`.

2. From the **Select Configuration** drop-down list, select **CastorServer Instance**.
3. Select the **Performance Configuration** tab.
4. Make sure the **Use Machine Default Performance Configuration** check box is cleared so that your logging settings are not overridden by the default settings.
5. In the **Statistics** column, select the check boxes for the counters that you want to log to the statistics repository.
6. In the Statistics Properties group, in the **Logging Frequency (min)**, specify how often (in minutes) you want the performance counters to log information.
7. From the **Persist statistics** drop-down list, select **Yes**.
8. From the **File** menu, select **Save**. The changes that you have made to the logging properties are saved.

The statistics database

Intelligence Server logs the specified statistics to the staging tables in the statistics repository. For a detailed examination of the staging tables in the statistics repository, see the *Statistics Data Dictionary* in the *Supplemental Reference for System Administration*.

If you are using Enterprise Manager to monitor your statistics, the database that hosts the staging tables also contains the Enterprise Manager data warehouse. The information in the staging tables is processed and loaded into the data warehouse as part of the data load process. For information about the structure of the Enterprise Manager data warehouse, see the *Enterprise Manager Data Dictionary* in the Supplemental Reference for System Administration. For steps on configuring Enterprise Manager and scheduling data loads, see the Enterprise Manager chapter in the *MicroStrategy Operations Manager Guide*.

Intelligence Server may open up to one database connection for each project that is configured to log statistics. For example, in a project source with four projects, each of which is logging statistics, there may be up to four database connections opened for logging statistics. However, the maximum number of database connections is typically seen in high-concurrency environments.

In a clustered environment, each node of the cluster requires a database connection for each project loaded onto that node. For example, a two-node cluster with 10

 © 2016, MicroStrategy Inc.

projects loaded on each node has 20 connections to the warehouse (10 for each node). Even if the same 10 projects are loaded on both nodes, 20 database connections exist.

Supported database platforms

MicroStrategy supports the following database platforms for use with Intelligence Server statistics:

- SQL Server
- Oracle
- Teradata
- IBM DB2 UDB
- Sybase ASE

For information about the specific versions of each database that are supported, see the *MicroStrategy Readme*.

Logging all statistics from a project source to the same database

By default, all projects for a project source must be configured to log statistics individually. This configuration is called **Complete Session Logging**. It allows some projects to log statistics to a database and some projects to log to another database.

The Enterprise Manager data warehouse must be in the same database as the statistics repository for a project. If you are using Enterprise Manager in a complete session logging configuration, there are as many Enterprise Manager data warehouses as there are statistics repositories. A separate Enterprise Manager project must be configured for each statistics repository.

MicroStrategy recommends that you configure all projects in your project source to log statistics to the same database. This is accomplished by configuring your system to use **Single Instance Session Logging**. This can minimize session logging and optimize system performance.

Under single instance session logging, you must still specify which statistics are logged for each individual project in the project source, as described in *Overview of Intelligence Server statistics, page 288*.

To use single instance session logging successfully, the selected single instance session logging project must be loaded onto the Intelligence Server at startup. If clustered Intelligence Servers are being used, the project must be loaded onto all the clustered Intelligence Servers. Failing to load this project on all servers at startup results in a loss of session statistics for any Intelligence Server on which the project is not loaded at startup. For details on the possible side effects of not loading all projects, see MicroStrategy Tech Note TN14591.

© 2016, MicroStrategy Inc.

To log all statistics from a project source to the same database

1. In Developer, right-click the project source and select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
2. On the left, expand **Statistics**, then select **General**.
3. Select the **Single Instance Session Logging** option.
4. Select a project from the drop-down list. The statistics for all projects on this Intelligence Server are logged to the database instance specified for this project.
5. Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.

Best practices for recording Intelligence Server statistics

MicroStrategy recommends the following best practices for logging Intelligence Server statistics:

- Configure your system for single instance session logging, so that all projects for a project source use the same statistics repository. This can reduce duplication, minimize database write time, and improve performance. For information about single instance session logging, see *Logging all statistics from a project source to the same database, page 291*.
- Use the sizing guidelines (see *Sizing guidelines for the statistics repository, page 292*) to plan how much hard disk space you need for the statistics repository.
- Use Enterprise Manager to monitor and analyze the statistics information. For more information about Enterprise Manager, see the Enterprise Manager chapter in the *MicroStrategy Operations Manager Guide*.

Sizing guidelines for the statistics repository

The following guidelines can help you determine how much space you need for the statistics repository. These guidelines are for planning purposes; MicroStrategy recommends that you monitor the size of your statistics repository and adjust your hardware requirements accordingly.

- When the Basic Statistics, Report Job Steps, Document Job Steps, Report SQL, Report Job Tables/Columns Accessed, and Prompt Answers statistics are logged, a user executing a report increases the statistics database size by an average of 70 kilobytes.

  This value assumes that large and complex reports are run as often as small reports. In contrast, in an environment where more than 85 percent of the reports that are executed return fewer than 1,000 cells, the average report increases the statistics database size by less than 10 kilobytes.

 © 2016, MicroStrategy Inc.

- When the Subscription Deliveries and Inbox Messages statistics are logged, each subscription that is delivered increases the statistics database size by less than 100 kilobytes. This is in addition to the database increase from logging the report execution.
- When performance counters are logged to the statistics database, each performance counter value that is logged increases the database size by an average of 0.4 kilobyte. You can control this table's growth by specifying what counters to log and how often to log each. For more information on logging performance counters to the statistics database, including instructions, see *Recording performance counters in the statistics tables, page 289*.

To determine how large a database you need, multiply the space required for a report by the number of reports that will be run over the amount of time you are keeping statistics. For example, you may plan to keep the statistics database current for six months and archive and purge statistics data that are older than six months. You expect users to run an average of 400 reports per day, of which 250, or 63 percent, return fewer than 1,000 rows, so you assume that each report will increase the statistics table by about 25 kilobytes.

```
25 KB/report * 400 reports/day * 30 days/month * 6 months
= 1,800,000 KB or 1.8 GB
```

According to these usage assumptions, you decide to allocate 2 GB of disk space for the statistics database.

Configuring Intelligence Server to log statistics

Below is a high-level overview of the steps to configure a project to log statistics.

1. Create the statistics database (see *Creating the statistics database, page 293*).
2. Create the statistics tables in the statistics database (see *Creating statistics tables in the statistics database, page 294*).
3. Configure your project to log statistics to the specified database (see *Setting the statistics database instance for a project, page 295*).
4. Choose what statistics to log from that project (see *Specifying which statistics to log, page 298*).

Creating the statistics database

You can store Intelligence Server statistics in an existing database in your system, or create a new database.

Do not store the statistics in the same database that you are using for either your MicroStrategy metadata or your data warehouse.

- To use an existing database, note its Data Source Name (DSN). This DSN is used when you create the statistics tables.

If you choose to use Enterprise Manager to analyze the statistics, this DSN is also used to specify the data warehouse location for Enterprise Manager. For information on Enterprise Manager, see the Enterprise Manager chapter in the *MicroStrategy Operations Manager Guide*.

- To create a new database, follow the procedure below. For a list of databases that are certified for use with Intelligence Server statistics, see *Supported database platforms, page 291*, or see the *MicroStrategy Readme*.

To create a new statistics database

1. Create the empty data warehouse database. (This is generally performed by your database administrator.) This database must be one of the databases certified for Intelligence Server statistics, as listed in the *MicroStrategy Readme*.
2. Use the MicroStrategy Connectivity Wizard to create a Data Source Name for the data warehouse. Note this DSN for later.

   To access the Connectivity Wizard, from the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and select **Connectivity Wizard**. For detailed instructions on using the Connectivity Wizard, see the Installation and Configuration Guide.

To avoid a situation in which some statistics database entries reports have incomplete information, synchronize the time of the Intelligence Server machine with the database time, if possible.

Creating statistics tables in the statistics database

After the statistics database has been created, or you have noted your existing database's DSN, you need to create the empty statistics tables for Intelligence Server to use. The MicroStrategy Configuration Wizard walks you through this process.

To create the empty statistics tables

1. Start the MicroStrategy Configuration Wizard.
   - Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Configuration Wizard**.
   - UNIX/Linux: Browse to the directory specified as the home directory during MicroStrategy installation, then browse to the folder `bin` and type `./mstrcfgwiz` and press ENTER.

 © 2016, MicroStrategy Inc.

2. On the Welcome page, select **Create Metadata, History List and Enterprise Manager Repositories** and click **Next**. The Repository Configuration: Repository Types page opens.
3. Select the **Statistics & Enterprise Manager** option and clear the other options. Click **Next**. The Repository Configuration: Statistics and Enterprise Manager Repository page opens.
4. From the **DSN** drop-down list, select the Data Source Name for the database that will contain your Enterprise Manager repository (the same database that you will use to log Intelligence Server statistics).

   Any table in this database that has the same name as a MicroStrategy statistics table is dropped. For a list of the MicroStrategy statistics tables, see the *Statistics Data Dictionary* in the Supplemental Reference for System Administration.

5. In the **User Name** and **Password** fields, enter a valid login and password for the data warehouse database.

   The user name you specify must have permission to create and drop tables in the database, and permission to create views.

6. If you want to use a custom SQL script for creating the repository, click **Advanced**.
   - In the **Script** field, the default script file name is displayed. The selected script depends on the database type that you specified earlier.
   - To select a different script, click **...** (the **Browse** button) to browse to and select a script that corresponds to the DBMS for the repository.
7. Click **Next**. The Configuration Wizard connects to the database.

   If Enterprise Manager statistics tables already exist in this database, it prompts you for whether to re-create the tables. To re-create them, click **Yes**. To leave the existing tables in place, click **No**.

   Clicking **Yes** deletes the existing tables and all information in them.

8. The Summary page lists the tasks that will be performed. To create the tables, click **Finish**. The process can take several minutes.

Setting the statistics database instance for a project

Once the statistics repository has been created, you must configure your project to log statistics to this database.

MicroStrategy recommends that you configure your system to use single instance session logging. In this configuration, statistics for all projects in a project source are logged to a single database. To enable single instance session logging, in the Intelligence Server Configuration Editor, in the **Statistics: General** category, select **Single Instance Session Logging** and, from the drop-down list, select a project. Then specify that project's statistics database using the procedure below. For steps on enabling single instance

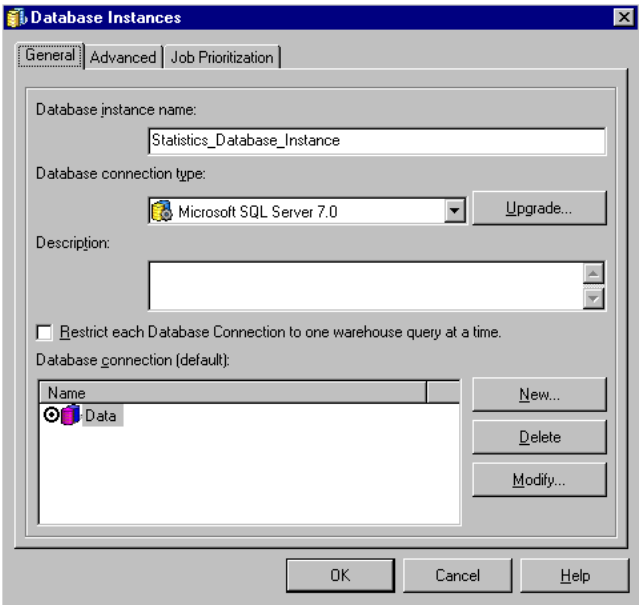
session logging, see *Logging all statistics from a project source to the same database, page 291*.

To set up a project to log statistics

1. In Developer, log in to the server (three-tier) project source containing the projects for which you want to log statistics. You must log in as a user with the Configure Server Basic privilege.
2. Right-click the project that you want to monitor and select **Project Configuration**. The Project Configuration Editor opens.

   If you are using single instance session logging, the project that you select to configure must be the project that you selected when you set up single instance session logging.

3. Expand the **Database Instances** category, and select the **SQL Data warehouses** subcategory.
4. You need to create a new database instance for the statistics repository database. Click **New**. The Database Instances dialog box opens.



5. In the **Database instance name** field, type in a name for the statistics repository database instance.
6. From the **Database connection type** drop-down list, select the database type and version that corresponds to the statistics repository database DBMS.
7. You need to create a new database connection to connect to the database instance. Click **New**. The Database Connections dialog box opens.
8. In the **Database connection name** field, type a name for the database connection.

 © 2016, MicroStrategy Inc.

9 From the **ODBC Data Sources** list, select the Data Source Name used to connect to the statistics repository database.

10 Enable parameterized queries in the statistics repository database connection. To do this, on the **Advanced** tab, select the **Use parameterized queries** check box.

11 You need to create a new database login to log in to the database instance. On the **General** tab, click **New**. The Database Logins dialog box opens.

12 Type a name for the new database login in the **Database login** field.

If this database login is more than 32 characters long, the statistics logging will generate errors in the DSS Errors log.

13 Type a valid database login ID and password in the corresponding fields.

MicroStrategy does not validate this login ID and password, so be careful to type them correctly.

14 Click **OK** three times to return to the Project Configuration Editor. In each case before clicking **OK**, make sure your new database login and database connection are selected.

15 In the **Database Instances** category, select the **Statistics** subcategory.

16 From the **Statistics database instance** drop-down list, select your new statistics database instance.

17 Click **OK**. Your selections are saved, and the Project Configuration Editor closes.

Configure an additional database driver setting

If your statistics and Enterprise Manager repository is in an Oracle, Sybase, or Teradata database, you must configure an additional ODBC driver setting so the information is recorded properly in the statistics repository.

18 Open the ODBC Data Source Administrator tool in Windows.

19 Select the DSN for your statistics and Enterprise Manager repository and click **Modify**. The ODBC Driver Setup dialog box opens.

20 Perform the following according to your database:

- Oracle: click the **Advanced** tab and select the **Enable SQLDescribeParam** check box.
- Sybase: click the **Advanced** tab and select the **Enable Describe Parameter** check box.
- Teradata: click **Options** and select the **Enable Extended Statement Information** check box.

21 Click **OK** twice to save the change and close the ODBC Data Source Administrator dialog box.

© 2016, MicroStrategy Inc.

Specifying which statistics to log

Once you have specified a statistics database instance for a project, you can select what statistics to log. For detailed information about what statistics can be logged, see *Overview of Intelligence Server statistics, page 288*.

You must specify what statistics to log for all projects that log statistics. Single instance session logging (see *Logging all statistics from a project source to the same database, page 291*) causes all projects on a project source to share the same statistics database, but not to log the same statistics.

To log information from performance counters, use the Diagnostics and Performance Logging Tool. For steps on how to log performance information, see *Recording performance counters in the statistics tables, page 289*.

To specify which statistics to log

1. In Developer, log in to the project source containing the project for which you want to log statistics. You must log in as a user with the Configure Server Basic privilege.
2. Right-click the project that you want to monitor and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand the **Statistics** category, and select the **General** subcategory.
4. Select the **Basic Statistics** check box.
5. To log advanced statistics, select the check boxes for the statistics you wish to log. For information about each check box, see *Overview of Intelligence Server statistics, page 288*.
6. Click **OK**. The Project Configuration Editor closes and your selections are saved.
7. To begin logging statistics, unload and reload the project for which you are logging statistics:
    a. In Developer, expand **Administration**, then expand **System Administration**, then select **Project**. A list of all projects on this project source is displayed.
    b. Right-click the project, point to **Administer Project**, and select **Unload**. The project is unloaded.
    c. Right-click the project, point to **Administer Project**, and select **Load**. The project is reloaded and configured to log statistics.

 © 2016, MicroStrategy Inc.

Additional monitoring tools

In addition to the logging tools and system monitors listed above, MicroStrategy provides several tools that help you track system usage and changes to the system. An overview of each tool is given below, along with a location for more information.

Diagnostics and Performance Logging Tool

MicroStrategy log files are records of all low-level system activities. You can configure Intelligence Server to record diagnostics information at various levels. You can also log performance information, such as the time taken to perform various operations and the total number of operations performed.

You manage log files through the MicroStrategy Diagnostics and Performance Logging tool. For information about this tool, including instructions, see *Finding trouble spots using diagnostics, page 731*.

Integrity Manager

MicroStrategy Integrity Manager can compare reports and documents between two projects. This can help you determine how specific changes in a project environment, such as the regular maintenance changes to metadata objects or hardware and software upgrades, affect the reports and documents in that project. Integrity Manager can also provide performance data for reports and documents, by recording the amount of time each report or document takes to execute.

For more information about Integrity Manager, see *Chapter 16, Verifying Reports and Documents with Integrity Manager*.

Enterprise Manager

MicroStrategy Enterprise Manager helps you analyze Intelligence Server statistics. Enterprise Manager provides a prebuilt MicroStrategy project with more than a hundred reports and dashboards covering all aspects of Intelligence Server operation. You can also use Enterprise Manager's prebuilt facts and attributes to create your own reports so you can have immediate access to the performance and system usage information.

For steps on setting up Enterprise Manager and using the reports in it, see the Enterprise Manager chapter in the *MicroStrategy Operations Manager Guide*.

Health Center

MicroStrategy Health Center can help you prevent, diagnose, and fix problems in your MicroStrategy system. It can scan your system for problems, recommend an immediate solution to many of them, and prepare a diagnostic package of log files and other

© 2016, MicroStrategy Inc.

relevant system information to be sent to MicroStrategy Technical Support if necessary. Health Center can also view the log files generated with the Diagnostics and Performance Logging Tool.

For more information about Health Center, see *Chapter 17, Maintaining Your MicroStrategy System with Health Center*.

Windows Performance Monitor

The Windows Performance Monitor is not part of MicroStrategy, but it can be useful for monitoring your system resources. For information about using the Windows Performance Monitor to monitor how MicroStrategy system components are using system resources, see *Managing system memory and resources: Windows Performance Monitor, page 311*.

 © 2016, MicroStrategy Inc.

8

TUNING YOUR SYSTEM FOR BEST PERFORMANCE

Introduction

Tuning a MicroStrategy system is not an exact science. Because your system resources, application performance, and user requirements and expectations are unique, it is not possible for MicroStrategy to include an exact methodology or set of recommendations for optimization.

One of your most important jobs as a MicroStrategy system administrator is to find the balance that maximizes the use of your system's capacity to provide the best performance possible for the required number of users. This chapter discusses how to analyze your users' requirements, and the ways you can configure and tune your system to meet those requirements.

The topics covered in this section include:

- *Tuning overview and best practices, page 302*
- *Designing system architecture, page 306*
- *Managing system resources, page 310*
- *Managing user sessions, page 324*
- *Governing requests, page 331*
- *Managing job execution, page 336*
- *Governing results delivery, page 347*
- *Tuning your system for in-memory datasets*

© 2016, MicroStrategy Inc.

- *Designing reports, page 353*
- *Configuring Intelligence Server and projects, page 355*
- *Tuning Narrowcast Server and Intelligence Server, page 367*

Tuning overview and best practices

To get the best performance out of your MicroStrategy system, you must be familiar with the characteristics of your system and how it performs under different conditions. In addition to this, you need a plan for tuning the system. For example, you should have a base record of certain key configuration settings and performance measures, such as Enterprise Manager reports or diagnostics logs, before you begin experimenting with those settings. Make one change at a time and test the system performance. Compare the new performance to the base and see if it improved. If it did not improve, change the setting back to its previous value. This way, when system performance improves, you know which change is responsible.

The specifications of the machines that you use to run Intelligence Server, how you tune those machines, and how they are used depend on the number of users, number of concurrently active users, their usage patterns, and so on. MicroStrategy provides up-to-date recommendations for these areas on the MicroStrategy Knowledge Base.

As a high-level overview of tuning the system, you should first define your system requirements, and then configure the system's design using those requirements. The following topics lay the foundation for the specific tuning guidelines that make up the rest of this chapter.

- *Defining the system requirements, page 302*
- *Configuring the system design, page 303*
- *Best practices for tuning your system, page 305*

Defining the system requirements

You most likely have certain expectations or requirements that the system must meet for it to be considered a success. For example, you may have a set of requirements similar to one of these scenarios:

- Global Web-based deployment for 400 users with 15-second response time for prompted reports and the ability to subscribe to personalized weekly sales reports.
- Internal deployment for 200 market research analysts accessing an enterprise data warehouse on a completely ad hoc basis.
- Web-based deployment for 1,500 remote users with access to pre-defined daily sales and inventory reports with 5-second response time.

 © 2016, MicroStrategy Inc.

These scenarios share common requirements that can help you define your own expectations for the system, such as the following:

- You may require that the system be able to handle a certain **number of concurrent users** logged in, or a certain **number of active users** running reports and otherwise interacting with the system.
- You may require a certain **level of performance**, such as report results returning to the users within a certain time, or that the results of report manipulation happen quickly, or that a certain number of reports can be run within an hour or within a day.
- You may require that users have access to certain **features**, such as scheduling a report for later execution, or sending a report to someone else via email, or that your users will be able to access their reports online through MicroStrategy Web.
- You may require that certain **functionality** be available in the system, such as allowing report flexibility so users can run ad hoc, predefined, prompted, page-by, or Intelligent Cube reports.

Configuring the system design

It is important to understand that the MicroStrategy business intelligence system has a limited capacity. It cannot serve an unlimited number of users and process an unlimited number of jobs in a short time. This capacity can be thought of as a box shared by the two important goals of serving the necessary number of user sessions (through which users **submit requests**) and maximizing the number of jobs executed (which **return results**).

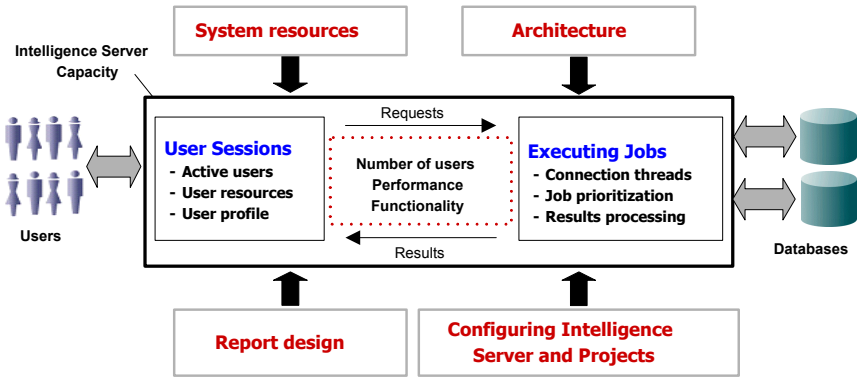
The limits that the system encounters may be Intelligence Server machine capacity, the data warehouse's throughput capacity, or the network's capacity.

The main factors that affect the system's capacity are:

- The **system resources** available (including memory)
- The **architecture** of the system and network
- The **design of the reports** that are executed
- The **configuration of Intelligence Server and projects** to determine how system resources can be used

The diagram below illustrates these factors that influence the system's capacity.

© 2016, MicroStrategy Inc.



UNIX and Linux systems allow processes and applications to run in a virtual environment. Intelligence Server Universal installs on UNIX and Linux systems with the required environment variables set to ensure that the server's jobs are processed correctly. However, you can tune these system settings to fit your system requirements and improve performance. For more information, see the *Planning Your Installation* chapter of the *MicroStrategy Installation and Configuration Guide*.

Configuring run-time capacity variables

Run-time capacity variables are factors that influence performance and capacity after Intelligence Server has started. The two run-time capacity variables are user sessions (see *Managing user sessions, page 324*) and executing jobs (see *Managing job execution, page 336*).

These runtime capacity variables are interrelated with system capacity. If you change settings in one, the others are affected. For example, if you place more emphasis on serving more user sessions, the job execution may suffer because it does not have as much of the system capacity available to use. Or, for example, if you increase Intelligence Server's capacity, it could execute jobs more quickly or it could serve more user sessions.

Accessing the system configuration editors

Many of the options in the following sections are specified in the Intelligence Server Configuration Editor or the Project Configuration Editor.

Prerequisites

- You must have the Configure Governing privilege for the project or project source.
- You must have Configuration permissions for the server object. In addition, to access the Project Configuration Editor you must have Write permission for the project object. For more information about server object permissions, see *Permissions for server governing and configuration, page 62*.

 © 2016, MicroStrategy Inc.

To access the Intelligence Server Configuration Editor

1. In Developer, log into a project source.
2. From the **Administration** menu, point to **Server** and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.

To access the Project Configuration Editor for a project

1. In Developer, log into a project source.
2. Expand the project that you want to configure.
3. From the **Administration** menu, point to **Projects** and then select **Project Configuration**. The Project Configuration Editor opens.

Best practices for tuning your system

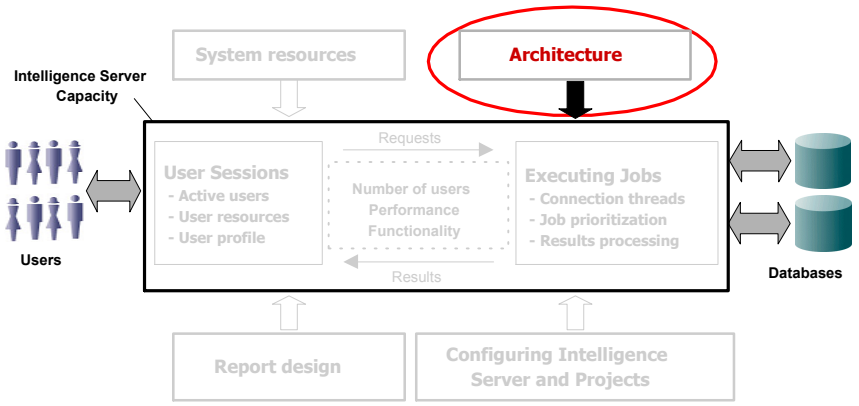
MicroStrategy recommends the following best practices for designing, configuring, and tuning your MicroStrategy system. For detailed information about increasing system performance by tuning the governing settings, see the remainder of this chapter.

- When designing your projects and data warehouse, follow the MicroStrategy best practices as outlined in the Project Design Guide.
- When configuring your network, follow the MicroStrategy best practices as outlined in *Network configuration best practices, page 308*.
- Use the Windows Performance Monitor to track the usage of system resources on Intelligence Server machines, as described in *Managing system memory and resources: Windows Performance Monitor, page 311*.
- Use Intelligence Server's Memory Contract Manager to manage memory usage, as described in *Governing Intelligence Server memory use with Memory Contract Manager, page 316*.
- Use MicroStrategy system privileges to restrict users' access to certain features, as described in *Governing user profiles, page 329*.
- Assign a high priority to more time-sensitive jobs, and a low priority to jobs that may use a great deal of system resources, as described in *Prioritizing jobs, page 340*.
- Enable Intelligence Server thread balancing, as described in *Intelligence Server thread balancing, page 343*.
- Ensure that report and document designers are aware of the features that can place an exceptionally heavy load on the system. These features are listed in detail in *Designing reports, page 353*.

© 2016, MicroStrategy Inc.

Designing system architecture

The choices that you make when designing the architecture of your MicroStrategy system have a significant impact on system performance and capacity.



Choices that you must make when designing your system architecture include:

- How the data warehouse is configured (see *How the data warehouse can affect performance, page 306*)
- The physical location of machines relative to each other and the amount of bandwidth between them (see *How the network can affect performance, page 307*)
- Whether you cluster several Intelligence Servers together and what benefits you can get from clustering (see *How clustering can affect performance, page 310*)

How the data warehouse can affect performance

The data warehouse is a crucial component of the business intelligence system. If it does not perform well, the entire system's performance suffers. The data warehouse platform or RDBMS and the data warehouse's design and tuning are factors that can affect your system's performance.

Platform considerations

The size and speed of the machines hosting your data warehouse and the database platform (RDBMS) running your data warehouse both affect the system's performance. A list of supported RDBMSs can be found in the *MicroStrategy Readme*. You should have an idea of the amount of data and the number of users that your system serves, and research which RDBMS can handle that type of load.

 © 2016, MicroStrategy Inc.

Design and tuning considerations

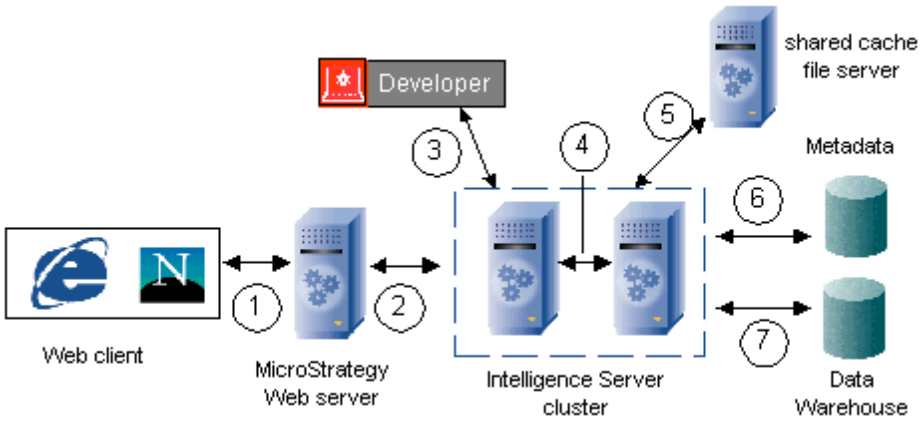
Your data warehouse's design (also called the physical warehouse schema) and tuning are important and unique to your organization. They also affect the performance of your business intelligence system. The discussion of the set of trade-offs that you must make when designing and tuning the data warehouse is out of the scope of this guide. Examples of the types of decisions that you must make include:

- Will you use a normalized, moderately normalized, or fully denormalized schema?
- What kind of lookup, relate, and fact tables will you need?
- What aggregate tables will you need?
- What tables do you need to partition and how?
- What tables will you index?

For more information about data warehouse design and data modeling, see the Advanced Reporting Guide and Project Design Guide.

How the network can affect performance

The various components of the MicroStrategy system need to be installed on different machines for performance reasons. The network plays an important role in connecting these components. In the diagram below, the separate components of the MicroStrategy system are linked by lines representing the network. The steps that occur over each connection are described in the table below the diagram.



Step	Protocol	Details
1	HTTP	HTML sent from Web server to client. Data size is small compared to other points because results have been incrementally fetched from Intelligence Server and HTML results do not contain any unnecessary information.

Step	Protocol	Details
2	TPC/IP	XML requests are sent to Intelligence Server. XML report results are incrementally fetched from Intelligence Server.
3	TCP/IP	Requests are sent to Intelligence Server. (No incremental fetch is used.)
4	TCP/IP	Broadcasts between all nodes of the cluster (if implemented): metadata changes, Inbox, report caches. Files containing cache and Inbox messages are exchanged between Intelligence Server nodes.
5	TCP/IP	Files containing cache and Inbox messages may also be exchanged between Intelligence Server nodes and a shared cache file server if implemented (see *Sharing result caches and Intelligent Cubes in a cluster, page 374*).
6	ODBC	Object requests and transactions to metadata. Request results are stored locally in Intelligence Server object cache.
7	ODBC	Complete result set is retrieved from database and stored in Intelligence Server memory and/or caches.

The maximum number of threads used in steps 2 and 3 can be controlled in the Intelligence Server Configuration Editor, in the **Server Definition: General** category, in the **Number of Network Threads** field. Depending on how your network is configured, one network thread may be sufficient to serve anywhere from 64 to 1028 user connections.

Network configuration best practices

The network configuration, that is, where the components are installed in relation to each other, can have a large effect on performance. For example, if the physical distance between Intelligence Server and the data warehouse is great, you may see poor performance due to network delays between the two machines.

MicroStrategy recommends the following best practices for network design:

- Place the Web server machines close to the Intelligence Server machines.
- Place Intelligence Server close to the both the data warehouse and the metadata repository.
- Dedicate a machine for the metadata repository.
- If you use Enterprise Manager, dedicate a machine for the Enterprise Manager database (statistics tables and data warehouse).
- If you have a clustered environment with a shared cache file server, place the shared cache file server close to the Intelligence Server machines.

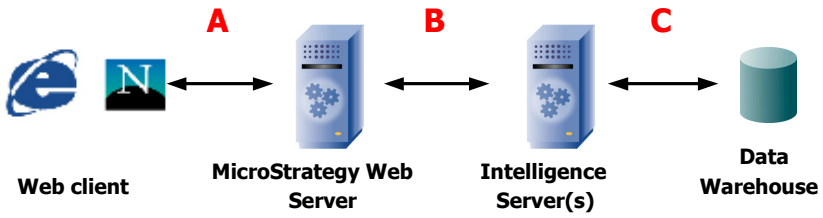
Network bandwidth and how its capacity is used

Your network design depends on the type of reports that your users typically run. These reports, in turn, determine the load they place on the system and how much network traffic occurs between the system components.

 © 2016, MicroStrategy Inc.

The ability of the network to quickly transport data between the components of the system greatly affects its performance. For large result sets, the highest load or the most traffic typically occurs between the data warehouse and the Intelligence Servers (indicated by **C** in the diagram below). The load between Intelligence Server and Web server is somewhat less (**B**), followed by the least load between the Web server and the Web browser (**A**).

This is illustrated in the diagram and explained below.



- Incremental fetch size directly influences the amount of traffic at A.
- Graphics increase network bandwidth at B.
- The load at C is determined primarily by the number of rows retrieved from the data warehouse. Actions such as sending SQL or retrieving objects from the metadata result in minimal traffic.
    - Cached reports do not cause any network traffic at C.
    - Report manipulations that do not cause SQL to be generated and sent to the data warehouse (such as pivot, sort, and page-by) are similar to running cached reports.
    - Report manipulations that cause SQL to be generated and sent to the data warehouse are similar to running non-cached reports of the same size.

After noting where the highest load is on your network, you can adjust your network bandwidth or change the placement of system components to improve the network's performance.

You can tell whether or not your network configuration has a negative effect on your system's performance by monitoring how much of your network's capacity is being used. Use the Windows Performance Monitor for the object **Network Interface**, and the watch the counter **Total bytes/sec** as a percent of your network's bandwidth. If it is consistently greater than 60 percent (for example), it may indicate that the network is negatively affecting the system's performance. You may wish to use a figure different than 60 percent for your system.

To calculate the network capacity utilization percent, take the total capacity, in terms of bits per second, and divide it by (Total bytes per second * 8). (Multiply the Total Bytes per second by 8 because 1 byte = 8 bits.)

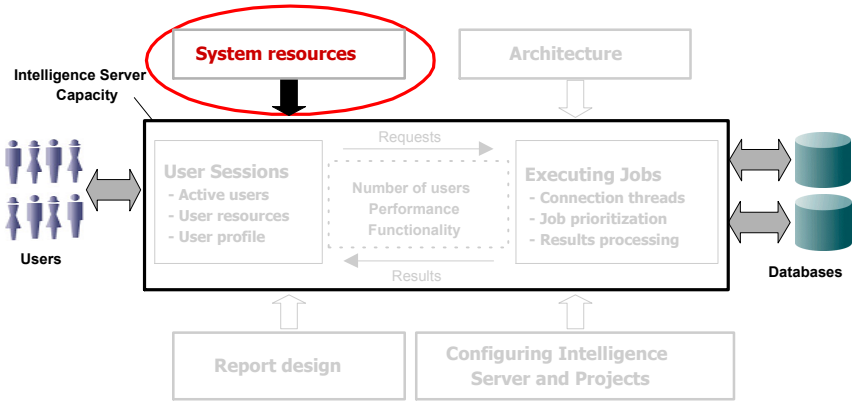
The **Current Bandwidth** counter in Performance Monitor gives an approximate value of total capacity because it is only an estimate. You may want to use another network monitoring utility such as `NetPerf` to get the actual bandwidth figure.

How clustering can affect performance

Clustering several Intelligence Server machines provides substantial gains in memory and CPU capacity because multiple machines are sharing the work. Clustering has additional benefits for your system as well. The clustering feature is built into Intelligence Server and is available out of the box if you have the proper license. For more information on clustering Intelligence Servers, including instructions, see *Chapter 9, Clustering Multiple MicroStrategy Servers*.

Managing system resources

If you had unlimited money, you could create a system that would impose few limits on system capacity. While system resources is not the place to save money when building a business intelligence system, you may not have the resources that you wish you could have.



You must make certain choices about how to maximize the use of your system's resources. Because Intelligence Server is the main component of the MicroStrategy system, it is important that the machines running it have sufficient resources for your needs. These resources include:

- The processors (*Processor type, speed, and number of processors, page 311*)
- Physical disk characteristics (*Physical disk, page 312*)
- The amount of memory (*Memory, page 312*)

The Installation and Configuration Guide contains detailed information about small, medium, and large configurations.

 © 2016, MicroStrategy Inc.

Managing system memory and resources: Windows Performance Monitor

Microsoft's Performance Monitor is a tool that is part of the Windows operating system. You can use it to monitor how memory and other resources are being used on a machine. You can also use it to access specific MicroStrategy performance counters.

For more information on Windows Performance monitor, see `http://www.microsoft.com/`.

To view the MicroStrategy-specific performance counters

1. In the Windows Performance Monitor, on the toolbar, click the **View Log Data** icon. The System Monitor Properties dialog box opens.
2. Select the **Data** tab.
3. Click **Add**. The Add Counters dialog box opens.
4. From the **Performance Object** drop-down list, select either **MicroStrategy Server Jobs** or **MicroStrategy Server Users**.
5. Select the desired counters from the list and click **Add**.
6. Click **Close**, then click **OK**. The dialog boxes close and the desired counters are now displayed in the Performance Monitor.

Processor type, speed, and number of processors

Intelligence Server recognizes the type and speed of the machine's CPUs, and performs faster on a machine with multiple CPUs. If Intelligence Server is consistently using a great deal of processor capacity, greater than 80 percent, for example, it may be a sign that a faster processor would improve the system's capacity. In Windows, you can monitor the processor usage with the Windows Performance Monitor. For instructions, see *Managing system memory and resources: Windows Performance Monitor, page 311*.

If you upgrade a machine's CPU, make sure you have the appropriate license to run Intelligence Server on the faster CPU. For example, if you upgrade the processor on the Intelligence Server machine from a 2 GHz to a 2.5 GHz processor, you should obtain a new license key from MicroStrategy.

Intelligence Server is also aware of the number of processors it is allowed to use according to the license key that you have purchased. For example, if a machine running Intelligence Server has two processors and you upgrade it to four, Intelligence Server uses only the two processors and ignores the additional two until you purchase a new license key from MicroStrategy. Also, if several Intelligence Server machines are clustered, the application ensures that the total number of processors being used does not exceed the number licensed.

For detailed information about CPU licensing, see *CPU licenses, page 205*.

© 2016, MicroStrategy Inc.

Physical disk

If the physical disk is used too much on a machine hosting Intelligence Server, it can indicate a bottleneck in the system's performance. To monitor physical disk usage in Windows, use the Windows Performance Monitor counters for the object **Physical Disk** and the counter **% Disk Time**. If the counter is greater than 80 percent on average, it may indicate that the machine does not have enough memory. This is because when the machine's physical RAM is full, the operating system starts swapping memory in and out of the page file on disk. This is not as efficient as using RAM. Therefore, Intelligence Server's performance may suffer.

By monitoring the disk utilization, you can see if the machine is consistently swapping at a high level. Defragmenting the physical disk may help lessen the amount of swapping. If that does not sufficiently lessen the utilization, consider increasing the amount of physical RAM in the machine. For information on how Intelligence Server uses memory, see *Memory, page 312*.

MicroStrategy recommends that you establish a benchmark or baseline of a machine's normal disk utilization, perhaps even before Intelligence Server is installed. This way you can determine whether or not Intelligence Server is responsible for excessive swapping because of limited RAM.

Another performance counter that you can use to gauge the disk's utilization is the **Current disk queue length**, which indicates how many requests are waiting at a time. MicroStrategy recommends using the % Disk Time and Current Disk Queue Length counters to monitor the disk utilization.

For instructions on how to use Windows Performance Monitor, see *Managing system memory and resources: Windows Performance Monitor, page 311*.

Memory

If the machine hosting Intelligence Server has too little memory, it may run slowly, or even shut down during memory-intensive operations. You can use the Windows Performance Monitor to monitor the available memory, and you can govern Intelligence Server's memory use with the Memory Contract Manager. The following topics can help you understand the best way to govern Intelligence Server's memory use:

- *Memory limitations: virtual memory, page 312*
- *Monitoring memory use with Performance Monitor, page 313*
- *How much memory does Intelligence Server use when it starts up?, page 314*
- *How does Intelligence Server use memory after it is running?, page 315*
- *Governing Intelligence Server memory use with Memory Contract Manager, page 316*

Memory limitations: virtual memory

The memory used by Intelligence Server is limited by the machine's virtual memory.

 © 2016, MicroStrategy Inc.

Virtual memory is the amount of physical memory (RAM) plus the Disk Page file (swap file). It is shared by all processes running on the machine, including the operating system.

When a machine runs out of virtual memory, processes on the machine are no longer able to process instructions and eventually the operating system may shut down. More virtual memory can be obtained by making sure that as few programs or services as possible are executing on the machine, or by increasing the amount of physical memory or the size of the page file.

Increasing the amount of virtual memory, and therefore the available private bytes, by increasing the page file size may have adverse effects on Intelligence Server performance because of increased swapping.

Private bytes are the bytes of virtual memory that are allocated to a process. Private bytes are so named because they cannot be shared with other processes: when a process such as Intelligence Server needs memory, it allocates an amount of virtual memory for its own use. The private bytes used by a process can be measured with the **Private Bytes** counter in the Windows Performance Monitor.

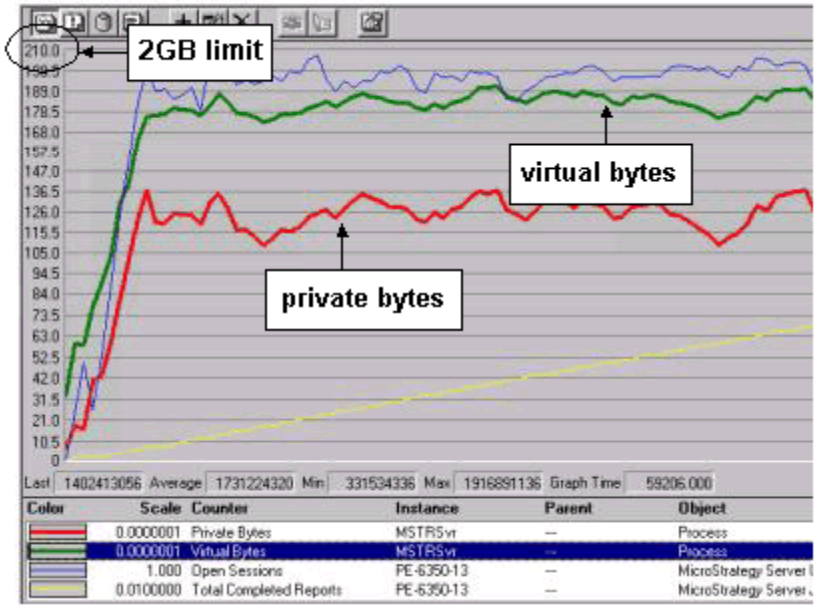
The governing settings built into Intelligence Server control its demand for private bytes by limiting the number and scale of operations which it may perform simultaneously. In most production environments, depletion of virtual memory through private bytes is not an issue with Intelligence Server.

Monitoring memory use with Performance Monitor

Microsoft's Performance Monitor is a useful tool for monitoring the amount of memory used on the entire machine or by a certain process (such as Intelligence Server). For instructions on how to use Performance Monitor to track Intelligence Server memory usage in your environment, see *Managing system memory and resources: Windows Performance Monitor, page 311*.

The two memory-related counters you should log with Performance Monitor are **Private Bytes** and **Virtual Bytes** for the Intelligence Server process (`Mstrsvr.exe`). A sample log of these two counters (along with others) for Intelligence Server is shown in the diagram below.

© 2016, MicroStrategy Inc.



The diagram above illustrates the gap between private bytes and virtual bytes in Intelligence Server. The **Virtual Bytes** counter represents memory that is reserved, not committed, for the process. **Private Bytes** represents memory actually being used by the process. Intelligence Server reserves regions of memory (called heaps) for use within the process. The heaps that are used by Intelligence Server cannot share reserved memory between themselves, causing the gap between reserved memory (virtual bytes) and memory being used by the process (private bytes) to increase further.

How much memory does Intelligence Server use when it starts up?

The amount of memory consumed during startup is affected by a number of factors such as metadata size, the number of projects, schema size, number of processing units, number of database connection threads required, and whether Intelligence Server is in a clustered configuration. Because these factors are generally static, the amount of memory consumed at startup is fairly constant. This lets you accurately estimate how much memory is available to users at runtime.

When Intelligence Server starts up, it uses memory in the following ways:

- It initializes all internal components and loads the static DLLs necessary for operation. This consumes 25 MB of private bytes and 110 MB of virtual bytes. You cannot control this memory usage.
- It loads all server definition settings and all configuration objects. This consumes an additional 10 MB of private bytes and an additional 40 MB of virtual bytes. This brings the total memory consumption at this point to 35 MB of private bytes and 150 MB of virtual bytes. You cannot control this memory usage.
- It loads the project schema (needed by the SQL engine component) into memory. The number and size of projects greatly impacts the amount of memory used. This consumes an amount of private bytes equal to three times the schema size and an amount of virtual bytes equal to four times the schema size. For example, with a schema size of 5 MB, the private bytes consumption would increase by 15 MB (3 * 5

 © 2016, MicroStrategy Inc.

MB). The virtual bytes consumption would increase by 20 MB (4 * 5 MB). You can control this memory usage by limiting the number of projects that load at startup time.

- It creates the database connection threads. This primarily affects virtual bytes consumption, with an increase of 1 MB per thread regardless of whether that thread is actually connected to the database. You cannot control this memory usage.

To calculate the amount of memory that Intelligence Server uses when it starts

If you are not performing this procedure in a production environment, make sure that you set all the configuration options as they exist in your production environment. Otherwise, the measurements will not reflect the actual production memory consumption.

1. Start Intelligence Server.
2. Once Intelligence Server has started, use Windows Performance Monitor to create and start a performance log that measures Private and Virtual bytes of the `MSTRSVR` process. For instructions on using the Windows Performance Monitor, see *Managing system memory and resources: Windows Performance Monitor, page 311*.
3. While logging with Performance Monitor, stop Intelligence Server. Performance Monitor continues to log information for the Intelligence Server process. You can confirm this by logging the counter information to the current activity window as well as the performance log.
4. Start Intelligence Server again. The amount of memory consumed should be easily measured.

How does Intelligence Server use memory after it is running?

Intelligence Server increases its memory use as needed during its operation. The following factors determine when memory use increases:

- Additional configuration objects: caching of user, connection map, and schedule and subscription information created or used after Intelligence Server has been started.
- Caches: result (report and document) caches, object caches, and element caches created after Intelligence Server has been started. The maximum amount of memory that Intelligence Server uses for result caches is configured at the project level. For more information about caches, see *Chapter 10, Improving Report and Document Response Time: Caching*.
- Intelligent Cubes: any Intelligent Cubes that have been loaded after Intelligence Server has been started. The maximum amount of memory used for Intelligent Cubes is configured at the project level. For details, see *Chapter 11, Managing Intelligent Cubes*.

© 2016, MicroStrategy Inc.

- User session-related resources: History List and Working set memory, which are greatly influenced by governing settings, report size, and report design. For details, see *Managing user sessions, page 324* and *Saving report results: History List, page 425*.
- Request and results processing: memory needed by Intelligence Server components to process requests and report results. This is primarily influenced by report size and report design with respect to analytical complexity. For details, see *Governing requests, page 331* and *Results processing, page 343*.
- Clustering: memory used by Intelligence Server to communicate with other cluster nodes and maintain synchronized report cache and History List information. For more information about clustering, see *Chapter 9, Clustering Multiple MicroStrategy Servers*.
- Scheduling: memory used by scheduler while executing reports for users when they are not logged in to the system. For more information about scheduling, see *Chapter 12, Scheduling Jobs and Administrative Tasks*.

Governing Intelligence Server memory use with Memory Contract Manager

Memory Contract Manager (MCM) is designed to protect Intelligence Server in cases where a memory request would cause the system to approach a state of memory depletion. When enabled, MCM grants or denies requests for memory from tasks in Intelligence Server. The requests are granted or denied according to user-configured limits on the amount of memory Intelligence Server is allowed to use. Because MCM is a component in Intelligence Server, it does not manage the actual memory used by Intelligence Server itself.

MCM governs the following types of requests:

- Database requests from either the MicroStrategy metadata or the data warehouse
- SQL generation
- Analytical Engine processing (subtotals, cross tabulation, analytic functions)
- Cache creation and updating
- Report parsing and serialization for network transfer
- XML generation

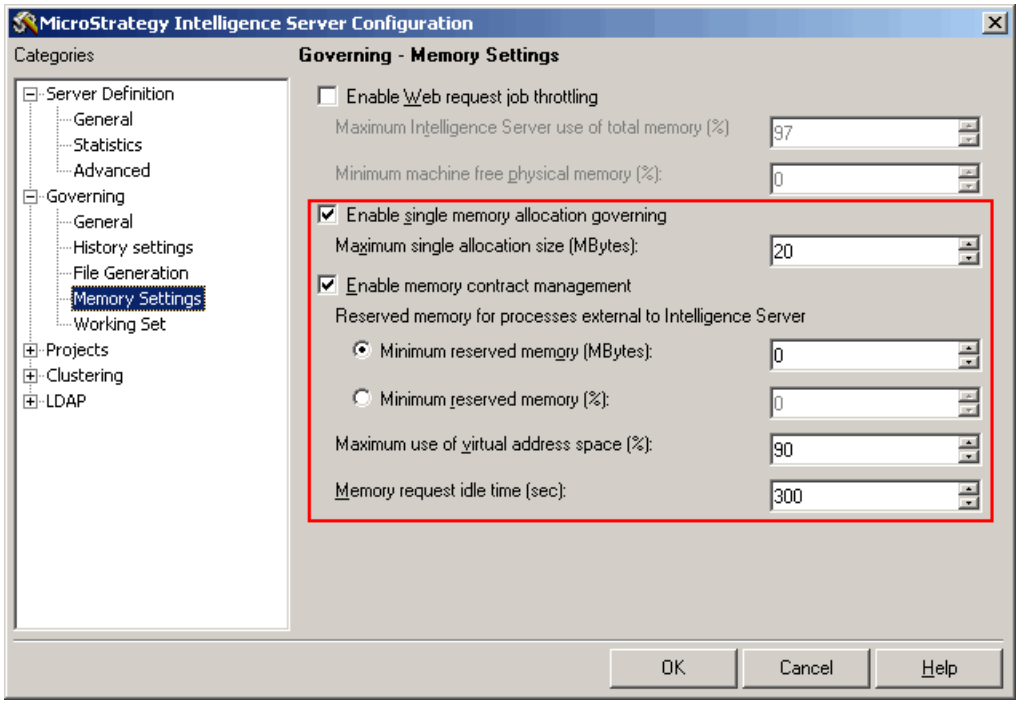
The memory load of the requests governed by MCM depends on the amount of data that is returned from the data warehouse. Therefore, this memory load cannot be predicted.

Requests such as graphing, cache lookup, or document generation use a predictable amount of memory and, thus, are not governed by MCM. For example, a request for a report returns an acceptable amount of data. A graph of the report's results would be based on the same data and, thus, would be allowed. Therefore, MCM is not involved in graphing requests. If the report was not returned because it exceeded memory limits, the graphing request would never be issued.

© 2016, MicroStrategy Inc.

Using the Memory Contract Manager

The MCM settings are in the Intelligence Server Configuration Editor, in the **Governing Rules: Default: Memory Settings** category.



The **Enable single memory allocation governing** option lets you specify how much memory can be reserved for a single Intelligence Server operation at a time. When this option is enabled, each memory request is compared to the **Maximum single allocation size (MBytes)** setting. If the request exceeds this limit, the request is denied. For example, if the allocation limit is set to 100 MB and a request is made for 120 MB, the request is denied, but a request for 90 MB is allowed.

If the Intelligence Server machine has additional software running on it, you may wish to set aside some memory for those processes to use. To reserve this memory, you can specify the **Minimum reserved memory** in terms of either the number of MB or the percent of total system memory. In this case, the total available memory is calculated as the initial size of the page file plus the RAM. It is possible that a machine has more virtual memory than MCM knows about if the maximum page file size is greater than the initial size.

Intelligence Server always reserves up to 500 MB for its own operation. If the machine does not have this much memory, or if the **Minimum reserved memory** would leave less than 500 MB available for Intelligence Server, no memory is reserved for other processes.

When MCM receives a request that would cause Intelligence Server's memory usage to exceed the **Minimum reserved memory** settings, it denies the request and goes into memory request idle mode. In this mode, MCM denies any requests that would deplete memory. MCM remains in memory request idle mode until the memory used by Intelligence Server falls below a certain limit, known as the low water mark. For

© 2016, MicroStrategy Inc.

information on how the low water mark is calculated, see *Memory water marks, page 319*. For information about how MCM handles memory request idle mode, see *Memory request idle mode, page 321*.

The **Maximum use of virtual address space** is applicable in 32-bit Windows operating systems. For 64-bit operating systems, use the **Minimum reserved memory** setting to control the amount of memory available for Intelligence Server.

The **Memory request idle time** is the longest time MCM remains in memory request idle mode. If the memory usage has not fallen below the low water mark by the end of the **Memory request idle time**, MCM restarts Intelligence Server. Setting the idle time to -1 causes Intelligence Server to remain idle until the memory usage falls below the low water mark.

How does MCM grant or deny a request?

When a task requests memory, it provides MCM with an estimate of how much memory it requires. If the request is granted, MCM decreases the amount of available memory and the task allocates memory from the memory subsystem. When the task is completed or canceled, the memory is released and the amount of available memory increases.

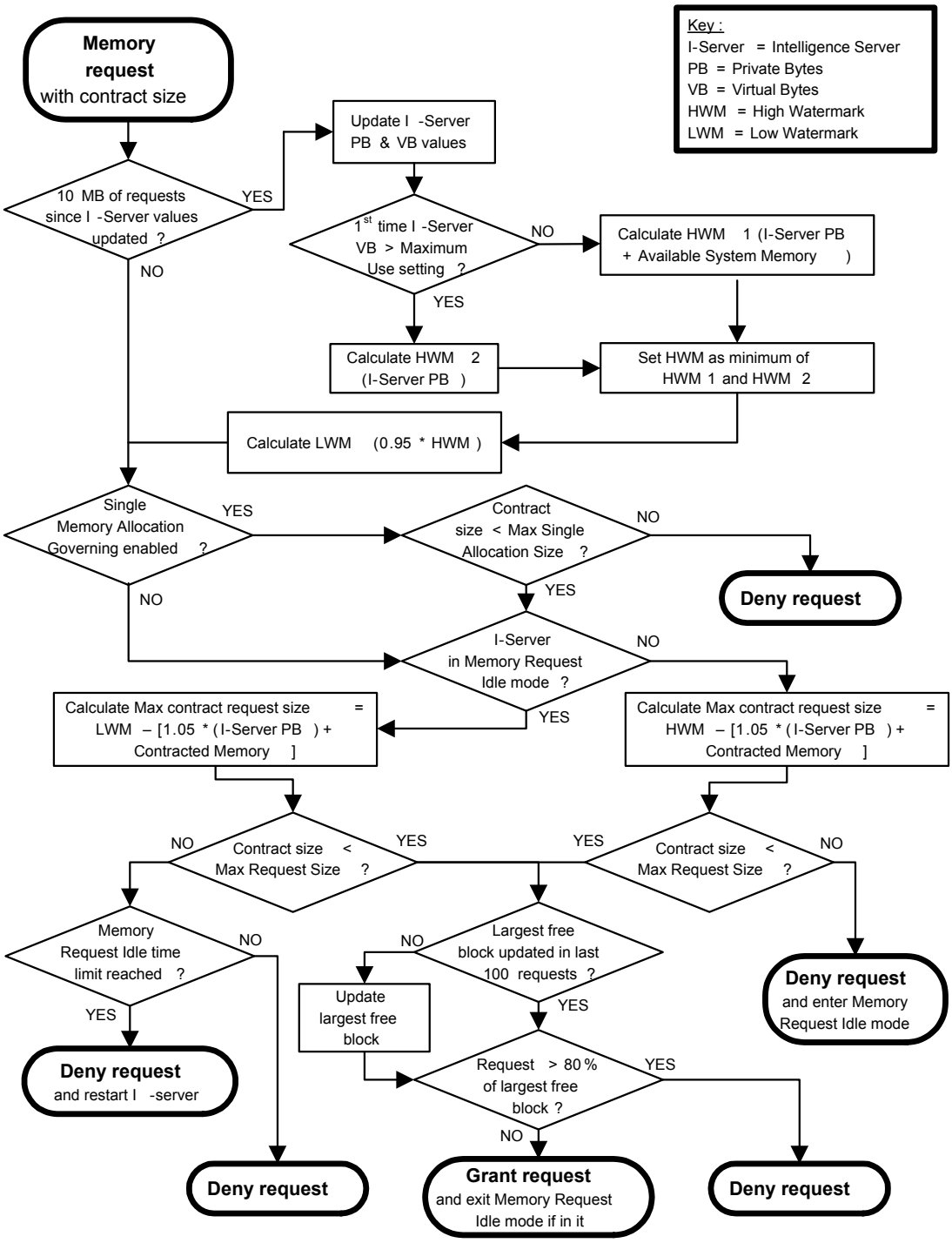
MCM does not submit memory allocations to the memory subsystem (such as a memory manager) on behalf of a task. Rather, it keeps a record of how much memory is available and how much memory has been contracted out to the tasks.

A memory request is granted if it meets the following criteria:

- It is smaller than the **Maximum single allocation size** setting.
- It is smaller than the high water mark, or the low water mark if Intelligence Server is in memory request idle mode. These water marks are derived from the Intelligence Server memory usage and the **Maximum use of virtual address space** and **Minimum reserved memory** settings. For detailed explanations of the memory water marks, see *Memory water marks, page 319*.
- It is smaller than 80 percent of the largest contiguous block of free memory to account for memory fragmentation.

To determine whether a memory request is granted or denied, MCM follows the logic in the flowchart below.

© 2016, MicroStrategy Inc.



Memory water marks

The high water mark (HWM) is the highest value that the sum of private bytes and outstanding memory contracts can reach before triggering memory request idle mode. The low water mark (LWM) is the value that Intelligence Server's private byte usage must drop to before MCM exits memory request idle mode. MCM recalculates the high and

low water marks after every 10 MB of memory requests. The 10 MB value is a built-in benchmark and cannot be changed.

Two possible values are calculated for the high water mark: one based on virtual memory, and one based on virtual bytes. For an explanation of the different types of memory, such as virtual bytes and private bytes, see *Memory, page 312*.

- The high water mark for virtual memory (HWM1 in the diagram above) is calculated as `(Intelligence Server private bytes + available system memory)`. It is recalculated for each potential memory depletion.

  The available system memory is calculated using the **Minimum reserved memory** limit if the actual memory used by other processes is less than this limit.

- The high water mark for virtual bytes (HWM2 in the diagram above) is calculated as `(Intelligence Server private bytes)`. It is calculated the first time the virtual byte usage exceeds the amount specified in the **Maximum use of virtual address space** or **Minimum Reserved Memory** settings. Because MCM ensures that Intelligence Server private byte usage cannot increase beyond the initial calculation, it is not recalculated until after Intelligence Server returns from the memory request idle state.

The high water mark used by MCM is the lower of these two values. This accounts for the scenario in which, after the virtual bytes HWM is calculated, Intelligence Server releases memory but other processes consume more available memory. This can cause a later calculation of the virtual memory HWM to be lower than the virtual bytes HWM.

The low water mark is calculated as 95 percent of the HWM. It is recalculated every time the HWM changes.

Memory contract management

Once the high and low water marks have been established, MCM checks to see if single memory allocation governing is enabled. If it is, and the request is for an amount of memory larger than the **Maximum single allocation size** setting, the request is denied.

If single memory allocation governing is not enabled, or if the request is for a block smaller than the **Maximum single allocation size** limit, MCM checks whether it is in memory request idle mode, and calculates the maximum contract request size accordingly:

- For normal Intelligence Server operation, the maximum request size is based on the high water mark. The formula is `[HWM - (1.05 *(Intelligence Server Private Bytes) + Outstanding Contracts)]`.
- In memory request idle mode, the maximum request size is based on the low water mark. The formula is `[LWM - (1.05 *(Intelligence Server Private Bytes) + Outstanding Contracts)]`.

The value of 1.05 is a built-in safety factor.

For normal Intelligence Server operation, if the request is larger than the maximum request size, MCM denies the request. It then enters memory request idle mode.

© 2016, MicroStrategy Inc.

If MCM is already in memory request idle mode and the request is larger than the maximum request size, MCM denies the request. It then checks whether the memory request idle time has been exceeded, and if so, it restarts Intelligence Server. For a detailed explanation of memory request idle mode, see *Memory request idle mode, page 321*.

If the request is smaller than the maximum request size, MCM performs a final check to account for potential fragmentation of virtual address space. MCM checks whether its record of the largest free block of memory has been updated in the last 100 requests, and if not, updates the record with the size of the current largest free block. It then compares the request against the largest free block. If the request is more than 80 percent of the largest free block, the request is denied. Otherwise, the request is granted.

After granting a request, if MCM has been in memory request idle mode, it returns to normal operation.

Memory request idle mode

When MCM first denies a request, it enters memory request idle mode. In this mode, MCM denies all requests that would keep Intelligence Server's private byte usage above the low water mark. MCM remains in memory request idle mode until one of the following situations occurs:

- Intelligence Server's memory usage drops below the low water mark. In this case, MCM exits memory request idle mode and resumes normal operation.
- MCM has been in memory request idle mode for longer than the **Memory request idle time**. In this case, MCM restarts Intelligence Server. This frees up the memory that had been allocated to Intelligence Server tasks, and avoids memory depletion.

The **Memory request idle time** limit is not enforced via an internal clock or scheduler. Instead, after every denied request MCM checks how much time has passed since the memory request idle mode was triggered. If this time is more than the memory request idle time limit, Intelligence Server restarts.

This eliminates a potentially unnecessary Intelligence Server restart. For example, a memory request causes the request idle mode to be triggered, but then no more requests are submitted for some time. A scheduled check at the end of the **Memory request idle time** would restart Intelligence Server even though no new jobs are being submitted. However, because Intelligence Server is completing its existing contracts and releasing memory, it is possible that the next contract request submitted will be below the low water mark. In this case, MCM accepts the request and resumes normal operation, without having to restart Intelligence Server.

When MCM forces Intelligence Server to restart because of the **Memory request idle time** being exceeded, it also writes the contents of Intelligence Server's memory use to disk. This memory dump is saved in the file `MCMServerStallDump.dmp` in the Intelligence Server folder. By default, this folder is located at `C:\Program Files (x86)\MicroStrategy\Intelligence Server\`.

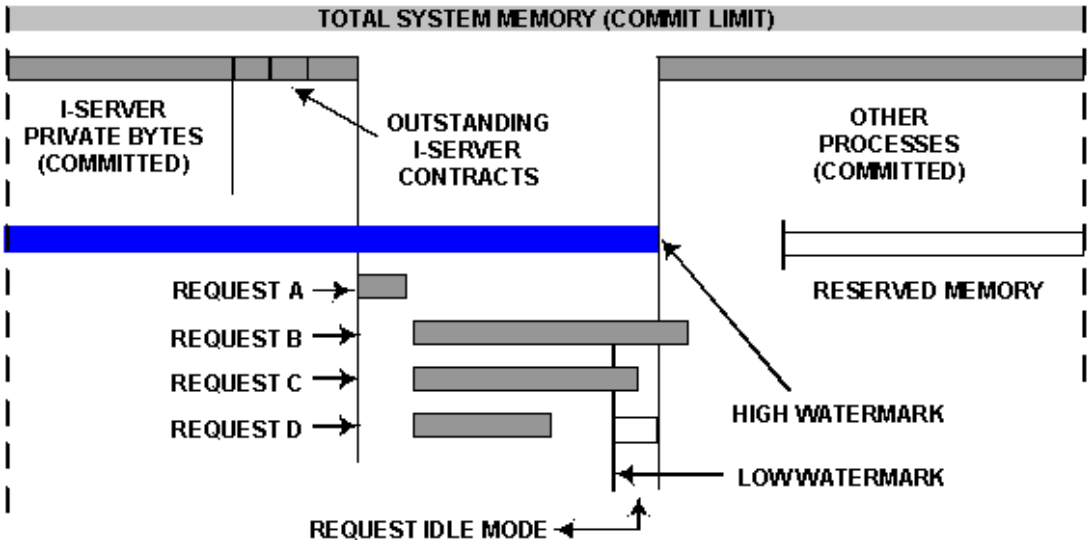
MicroStrategy recommends setting the **Memory request idle time** to slightly longer than the time it takes most large reports in your system to run. This way, Intelligence Server does not shut down needlessly while waiting for a task to complete. To help you

© 2016, MicroStrategy Inc.

determine the time limit, use Enterprise Manager to find out the average and maximum report execution times for your system. For instructions on using Enterprise Manager, see the *MicroStrategy Operations Manager Guide*.

System Memory Depletion

The diagram below shows an example of a potential depletion of system memory.



In this example, MCM grants memory request A. Once granted, a new memory contract is accounted for in the available system memory. Request B is then denied because it exceeds the high water mark, as derived from the **Maximum use of virtual address space** setting.

Once request B has been denied, Intelligence Server enters the memory request idle mode. In this mode of operation, it denies all requests that would push the total memory used above the low water mark.

In the example above, request C falls above the low water mark. Because Intelligence Server is in memory request idle mode, this request is denied unless Intelligence Server releases memory from elsewhere, such as other completed contracts.

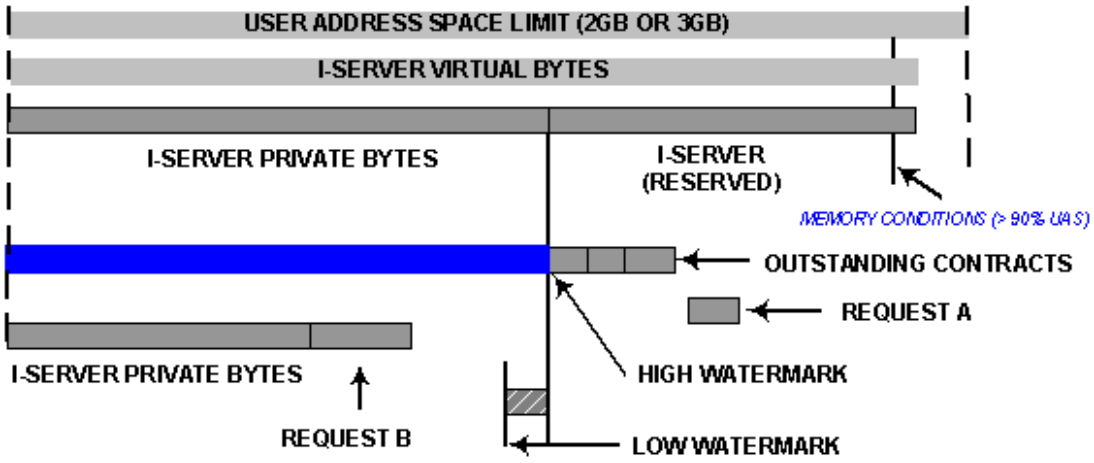
Request D is below the low water mark, so it is granted. Once it has been granted, Intelligence Server switches out of request idle mode and resumes normal operation.

If Intelligence Server continues receiving requests for memory above the low water mark before the **Memory request idle time** is exceeded, MCM shuts down and restarts Intelligence Server.

Virtual Byte Depletion

Below is a diagram of potential memory depletion due to available bytes in the Intelligence Server virtual address space.

© 2016, MicroStrategy Inc.



In this example, Intelligence Server has increased its private byte usage to the point that existing contracts are pushed above the high water mark. Request A is denied because the requested memory would further deplete Intelligence Server's virtual address space.

Once request A has been denied, Intelligence Server enters the memory request idle mode. In this mode of operation, all requests that would push the total memory used above the low water mark are denied.

The low water mark is 95 percent of the high water mark. In this scenario, the high water mark is the amount of Intelligence Server private bytes at the time when the memory depletion was first detected. Once the virtual byte high water mark has been set, it is not recalculated. Thus, for Intelligence Server to exit memory request idle mode, it must release some of the private bytes.

Although the virtual bytes high water mark is not recalculated, the virtual memory high water mark is recalculated after each request. MCM calculates the low water mark based on the lower of the virtual memory high water mark and the virtual bytes high water mark. This accounts for the scenario in which, after the virtual bytes high water mark is calculated, Intelligence Server releases memory but other processes consume more available memory. This can cause a later calculation of the virtual memory high water mark to be lower than the virtual bytes high water mark.

Intelligence Server remains in memory request idle mode until the memory usage looks like it does at the time of request B. The Intelligence Server private byte usage has dropped to the point where a request can be made that is below the low water mark. This request is granted, and MCM exits memory request idle mode.

If Intelligence Server does not free up enough memory to process request B before the **Memory request idle time** is exceeded, MCM restarts Intelligence Server.

Governing memory for requests from MicroStrategy Web products

You can limit the total amount of memory that Intelligence Server can use for serving requests from MicroStrategy Web, and you can set the amount of memory that must be

kept free for requests from MicroStrategy Web. These limits are enabled when the **Web Request job throttling** check box is selected. If either condition is met, all requests from MicroStrategy Web of any nature (log in, report execution, search, folder browsing) are denied until the conditions are resolved. For more details about each setting, see below.

- **Maximum Intelligence Server use of total memory** sets the maximum amount of total system memory (RAM plus Page File) that can be used by the Intelligence Server process compared to the total amount of memory on the machine.

  This setting is useful to prevent the system from servicing a Web request if memory is depleted. If the condition is met, Intelligence Server denies all requests from a MicroStrategy Web product or a client built with the MicroStrategy Web API.

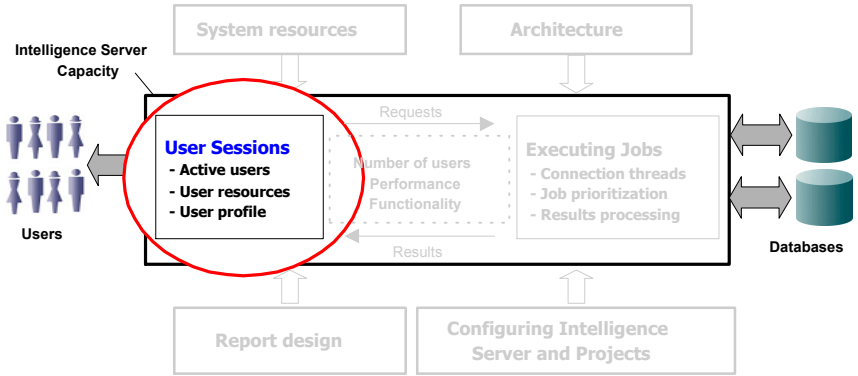
- **Minimum machine free physical memory** sets the minimum amount of RAM that must remain available for Web requests. This value is a percentage of the total amount of physical memory on the machine, not including the Page File memory.

  This can be useful if the machine is running applications other than Intelligence Server and you want to increase the chances that requests from MicroStrategy Web products are serviced using RAM and not the Page File, which does not work as efficiently.

Managing user sessions

Each user connection from a MicroStrategy client (MicroStrategy Web, Developer, Narrowcast Server, and others) establishes a user session on Intelligence Server. Each user session consumes a set amount of resources on the Intelligence Server machine and can consume additional resources depending on the actions that the user takes while she is connected.

The number of active users in a system (those actually executing reports and using the system) is considered a different category of user from concurrent users (those simply logged in).



This section covers:

 © 2016, MicroStrategy Inc.

- How the concurrent users and user sessions on your system use system resources just by logging in to the system (see *Governing concurrent users, page 325*)
- How memory and CPU are used by active users when they execute jobs, run reports, and make requests, and how you can govern those requests (see *Governing user resources, page 327*)
- How user profiles can determine what users are able to do when they are logged in to the system, and how you can govern those profiles (see *Governing user profiles, page 329*)

With the User Connection Monitor, you can track the users who are connected to the system. For details about how to use this system monitor, see *Monitoring users' connections to projects, page 57*.

Governing concurrent users

When a user logs in to a MicroStrategy system, a user session is established. This user session remains open until the user logs out of the system or the system logs the user out. Users that are logged in but are not doing anything still consume some resources on Intelligence Server. The more user sessions that are allowed on Intelligence Server, the more load those users can put on the system because each session can run multiple jobs.

To help control the load that user sessions can put on the system, you can limit the number of concurrent user sessions allowed for each project and for Intelligence Server. Also, both Developer and MicroStrategy Web have session timeouts so that when users forget to log out, the system logs them out and their sessions do not unnecessarily use up Intelligence Server resources.

For example, a user logs in, runs a report, then leaves for lunch without logging out of the system. If Intelligence Server is serving the maximum number of user sessions and another user attempts to log in to the system, that user is not allowed to log in. You can set a time limit for the total duration of a user session, and you can limit how long a session remains open if it is inactive or not being used. In this case, if you set the inactive time limit to 15 minutes, the person who left for lunch has her session ended by Intelligence Server. After that, another user can log in.

Intelligence Server does not end a user session until all the jobs submitted by that user have completed or timed out. This includes reports that are waiting for autoprompt answers. For example, if a MicroStrategy Web or Web Universal user runs a report with an autoprompt and, instead of answering the prompt, clicks the browser's Back button, an open job is created. If the user then closes his or her browser or logs out without canceling the job, the user session remains open until the open job "Waiting for Autoprompt" times out.

These user session limits are discussed below as they relate to software features and products.

Limiting the number of user sessions on Intelligence Server

This setting limits the number of user sessions that can be connected to an Intelligence Server. This includes connections made from MicroStrategy Web products, Developer, Distribution Services, Scheduler, or other applications that you may have created with the SDK. A single user account can establish multiple sessions on an Intelligence Server. Each session connects once to Intelligence Server and once to each project that the user accesses. In the User Connection Monitor, the connections made to Intelligence Server display as **<Server>** in the Project column. Project sessions are governed separately with a project level setting, **User sessions per project**, which is discussed below. When the maximum number of user sessions on Intelligence Server is reached, users cannot log in, except for the administrator, who can disconnect current users by means of the User Connection Monitor or increase this governing setting.

To specify this setting, in the Intelligence Server Configuration Editor, select the **Governing Rules: Default: General** category and type the number in the **Maximum number of user sessions** field.

Limiting user sessions per project

When a user accesses a project, a connection (called a user session) is established for the project and Intelligence Server. In the User Connection Monitor, the connections made to the project display the project name in the Project column. If you sort the list of connections by the Project column, you can see the total number of user sessions for each project.

You can limit the number of sessions that are allowed for each project. When the maximum number of user sessions for a project is reached, users cannot log in to the system. An exception is made for the system administrator, who can log in to disconnect current users by means of the User Connection Monitor or increase this governing setting.

To specify this setting, in the Project Configuration Editor for the project, select the **Governing Rules: Default: User sessions** category and type the number in the **User sessions per project** field.

You can also limit the number of concurrent sessions per user. This can be useful if one user account, such as “Guest,” is used for multiple connections. To specify this setting, in the Project Configuration Editor for the project, select the **Governing Rules: Default: User sessions** category and type the number in the **Concurrent interactive project sessions per user** field.

Limiting user session idle times

When a user logs in to Developer (in a three-tier configuration) or MicroStrategy Web, a user session is established. As long as the user logged into that session is using the project, creating or executing reports, and so on, the session is considered active. When the user stops actively using the session, this is considered idle time. You can specify the maximum amount of time a session can remain idle before Intelligence Server disconnects that session. This frees up the system resources that the idle session was using and allows other users to log in to the system if the maximum number of user sessions has been reached.

 © 2016, MicroStrategy Inc.

To specify this setting for Developer, in the Intelligence Server Configuration Editor, select the **Governing Rules: Default: General** category and, in the **User session idle time (sec)** field, type the number of seconds of idle time that you want to allow.

To specify this setting for MicroStrategy Web, in the Intelligence Server Configuration Editor, select the **Governing Rules: Default: General** category and, in the **Web user session idle time (sec)** field, type the number of seconds of idle time that you want to allow.

If designers are building Report Services documents and dashboards in MicroStrategy Web, set the **Web user session idle time (sec)** to 3600 to avoid a project source timeout.

Governing user resources

User sessions consume system resources when users log in to the system, especially when they use the History List and, in MicroStrategy Web, the Working Set. If a Web user's session expires and the system is configured to allow users to recover their session information, the stored session information uses resources. This section discusses these features and how you can govern them.

Like all requests, user resources are also governed by the Memory Contract Manager settings. For more information about Memory Contract Manager, see *Governing Intelligence Server memory use with Memory Contract Manager, page 316*.

History List

The History List is an in-memory message list that references reports that a user has executed or scheduled. The results are stored as History or Matching-History caches on Intelligence Server.

The History List can consume much of the system's resources. You can govern the resources used by old History List messages in the following ways:

- You can delete messages from the History List with a scheduled administrative task. For more information and instructions on scheduling this task, see *Scheduling administrative tasks, page 485*.
- In the Intelligence Server Configuration Editor, in the **History settings: General** category, you can limit the **Maximum number of messages per user**. If a user has hit this maximum and tries to add another message to the History List, the oldest message is automatically purged.
- In the Intelligence Server Configuration Editor, in the **History settings: General** category, you can set the **Message lifetime (days)**. Intelligence Server automatically deletes any History List messages that are older than the specified message lifetime.

For more information about the History List, including details on History List governing settings, see *Saving report results: History List, page 425*.

© 2016, MicroStrategy Inc.

Working set (MicroStrategy Web only)

When a user runs a report from MicroStrategy Web or Web Universal, the results from the report are added to the working set for that user's session and stored in memory on Intelligence Server. The working set is a collection of messages that reference in-memory report instances. A message is added to the working set when a user executes a report or retrieves a message from the History List. The purpose of the working set is to:

- Improve MicroStrategy Web performance for report manipulations, without having to run SQL against the data warehouse for each change
- Allow the efficient use of the web browser's Back button
- Allow users to manually add messages to the History List

Each message in the working set can store two versions of the report instance in memory: the **original version** and the **result version**. The original version of the report instance is created the first time the report is executed and is held in memory the entire time a message is part of the working set. The result version of the report instance is added to the working set only after the user manipulates the report. Each report manipulation adds what is called a delta XML to the report message. On each successive manipulation, a new delta XML is applied to the result version. When the user clicks the browser's Back button, previous delta XMLs are applied to the original report instance up to the state that the user is requesting. For example, if a user has made four manipulations, the report has four delta XMLs; when the user clicks the Back button, the three previous XMLs are applied to the original version.

Governing History List and Working set memory use in MicroStrategy Web

You can control the amount of the memory that is used by the History List and Working set in these ways:

- Limit the number of reports that a user can keep available for manipulation in a MicroStrategy Web product. This number is defined in the MicroStrategy Web products' interface in **Project defaults: History List** settings. You must select the **Manually** option for adding messages to the History List, then specify the number in the field labeled **If manually, how many of the most recently run reports and documents do you want to keep available for manipulation?** The default is 10 and the minimum is 1. The higher the number, the more memory the reports may consume. For details, see the MicroStrategy Web Help.
- Limit the maximum amount of RAM that all users can use for the working set. When the limit is reached and new report instances are created, the least recently used report instance is swapped to disk. To set this, in the Intelligence Server Configuration Editor, under the **Governing Rules: Default: Working Set** category, type the limit in the **Maximum RAM for Working Set cache (MB)** field.

Be aware of the following:

— If you set this limit to more memory than the operating system can make available, Intelligence Server uses a value of 100 MB. The maximum value for this setting is 65,536 MB (64 GB) on most operating systems. It is 2048 MB (2 GB) under Windows 2003.

 © 2016, MicroStrategy Inc.

— If you set this limit too low and you do not have enough hard disk space to handle the amount of disk swapping, reports may fail to execute in peak usage periods because the reports cannot write to memory or to disk.

If a user session has an open job, the user session remains open and that job's report instance is removed from the Working set when the job has finished or timed out. In this way, jobs can continue executing even after the user has logged out. This may cause excessive memory usage on Intelligence Server because the session's working set is held in memory until the session is closed. For instructions on how to set the timeout period for jobs, see *Limiting the maximum report execution time, page 334*.

Governing saved user session information (MicroStrategy Web only)

You can allow Web users to recover their document, report, or dashboard after their user session has been ended. If this feature is enabled and, for example, the user runs a report and walks away from his desk and the session times out, the user session information is saved. The next time the Web user logs in, if the recoverable session has not expired, the user can click a link to return to his recovered report. Enabling this feature uses disk space for storing the information. You can govern how long the sessions are stored before expiring. Long expiration times allow more information to be stored, thus using more system disk space. Shortening the expiration time more quickly frees up the system resources that the saved session was using.

To configure these settings, access the Intelligence Server Configuration Editor, select the **Governing Rules: Default: Temporary Storage Settings** category. To enable the feature, select the **Enable Web User Session Recovery on Logout** check box, and in the **Session Recovery backup expiration (hrs)** field, type the number of hours you want to allow a session to be stored. In **Session Recovery and Deferred Inbox storage directory**, specify the folder where the user session information is stored.

Governing user profiles

The user profile can be defined as what the user can do when logged in to the system. If you allow users to use certain features in the system, they can affect the system's performance. For example, when users schedule report executions, this creates user sessions on Intelligence Server, thus placing a load on it even when the users are not actively logged in.

You can limit these types of activities by restricting various privileges, as discussed below. For general information about privileges and the MicroStrategy security model, including instructions on how to grant and revoke privileges, see *Controlling access to functionality: Privileges, page 66*.

Subscription-related privileges

Allowing users to subscribe to reports to be run later can affect system performance. You can limit the use of subscriptions by using the Web Scheduled Reports and Schedule Request privileges.

© 2016, MicroStrategy Inc.

If you have Distribution Services or Narrowcast Server implemented in your system and users have the Web Scheduled Email or Web Send Now privileges, they can have a report emailed either at a set time or immediately. This causes the system to create a user session on Intelligence Server when the report is emailed.

For detailed information about subscribing to reports and documents, see *Scheduling reports and documents: Subscriptions, page 488*. For information about Distribution Services, see *Overview of Distribution Services, page 501*.

History List privileges

Allowing users to use the History List can consume extra system resources. Governing History List usage is discussed more fully in the previous section (see *History List, page 327*). The non-administrative privileges relating to the History List are:

- Web Subscribe To History List
- Web View History List
- Web Add To History List
- Use Link To History List in Email (Distribution Services)
- Use History List

Report manipulation privileges

The more manipulations that you allow users to do, the greater the potential for using more system resources. Manipulations that can use extra system resources include pivoting, page-by, and sorting. You can limit these manipulations with the following privileges:

- To limit the use of pivoting, use the Web Pivot Report and Pivot Report privileges.
- To limit the use of page-by, use the Web Switch Page-by Elements privilege.
- To limit the use of sorting, use the Web Sort and Modify Sorting privilege.

Exporting privileges

Exporting reports can consume large amounts of memory, especially when reports are exported to Excel with formatting. For more information on how to limit this memory usage, see *Limiting the number of XML cells, page 349*. The privileges related to exporting reports are found in the Common privilege group, and are as follows:

- Export to Excel
- Export to Flash
- Export to HTML
- Export to MicroStrategy File
- Export to PDF

 © 2016, MicroStrategy Inc.

- Export to Text

To restrict users from exporting any reports from MicroStrategy Web, use the Web Export privilege in the Web Reporter privilege group.

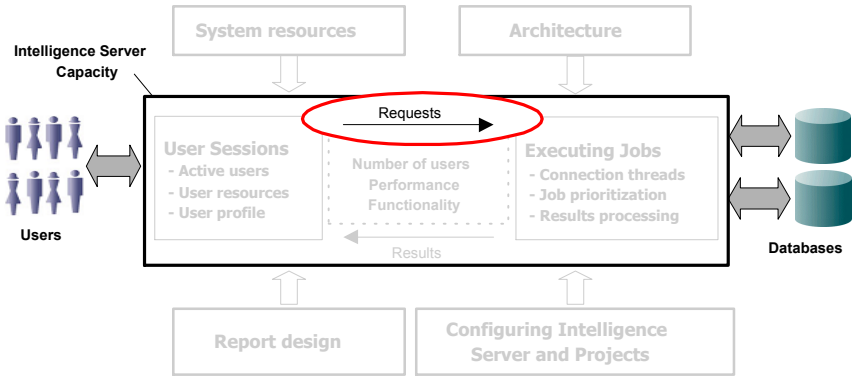
OLAP Services privileges

If you have purchased OLAP Services licenses for your users, they could use a great deal of the available system resources. For example, if your users are creating large Intelligent Cubes and doing many manipulations on them, the system will be loaded much more than if they are running occasional, small reports and not performing many manipulations.

The OLAP Services privileges are marked with a * in the list of all privileges (see the *List of Privileges* chapter in the Supplemental Reference for System Administration). For more details about how OLAP Services uses system resources, see *Intelligent Cubes, page 355*.

Governing requests

Each user session can execute multiple concurrent jobs or requests. This happens when users run documents that submit multiple child reports at a time or when they send a report to the History List, then execute another while the first one is still executing. Users can also log in to the system multiple times and run reports simultaneously. Again, this may use up a great deal of the available system resources.



To control the number of jobs that can be running at the same time, you can set limits on the requests that can be executed. You can limit the requests per user and per project. You can also choose to exclude reports submitted as part of a Report Services document from the job limits (see *Excluding document datasets from the job limits, page 332*).

Specifically, you can limit:

- The total number of jobs (*Limiting the total number of jobs, page 332*)

- The number of jobs per project (*Limiting the number of jobs per project, page 333*)
- The number of jobs per user account and per user session (*Limiting the number of jobs per user session and per user account, page 333*)
- The number of executing reports or data marts per user account (not counting element requests, metadata requests, and report manipulations) (*Limiting the number of executing jobs per user and project, page 334*)
- The amount of time reports can execute (*Limiting the maximum report execution time, page 334*)
- A report's SQL (per pass) including both its size and the time it executes (*Limiting a report's SQL per pass, page 335*)
- The amount of memory used for Intelligent Cubes (*Governing Intelligent Cube memory usage, page 465*)

Excluding document datasets from the job limits

Multiple jobs may be submitted when documents and reports are executed. For example, if you execute a document that has a prompt and three reports embedded in it, Intelligence Server processes five jobs: one for the document, one for the prompt, and three for the embedded dataset reports.

To avoid unexpectedly preventing document from executing, you can exclude report jobs submitted as part of document execution from the job limits. In this case, if you execute a document that has a prompt and three reports embedded in it, Intelligence Server would only count two jobs, the document and the prompt, towards the job limits described below.

To exclude document dataset jobs from the job limits, in the Intelligence Server Configuration Editor, select the **Governing Rules: Default: General** category, and select the **For Intelligence Server job and history list governing, exclude reports embedded in Report Services documents from the counts** check box. This selection applies to the project-level job limits as well as to the server-level limits.

Limiting the total number of jobs

You can limit the total number of concurrent jobs being processed by Intelligence Server. Concurrent jobs include report requests, element requests, and autoprompt requests that are executing or waiting to execute. Completed (open) jobs, cached jobs, or jobs that have returned an error are not counted. If the job limit is reached, a user sees an error message stating that the maximum number of jobs has been reached. The user needs to submit the job again.

To set this limit, in the Intelligence Server Configuration Editor, select the **Governing Rules: Default: General** category, and specify the value in the **Maximum number of jobs** field. You can also specify a maximum number of interactive jobs (jobs executed by a direct user request) and scheduled jobs (jobs executed by a scheduled request). A value of -1 indicates that there is no limit on the number of jobs that can be executed.

 © 2016, MicroStrategy Inc.

Limiting the number of jobs per project

You can limit the number of concurrent jobs that are being processed by Intelligence Server for a project. If you have multiple projects on an Intelligence Server, each can have its own job limit setting. Limiting the number of concurrent jobs per project helps reduce unnecessary strain on the system by limiting the amount of resources that concurrently executing jobs can take up.

Concurrent jobs include report requests, element requests, and autoprompt requests that are executing or waiting to execute. Finished jobs that are still open, cached jobs, and jobs that returned an error are not counted. If the limit is reached, a user sees an error message stating that the number of jobs per project is too high. The user then needs to submit the job again.

In a clustered system, these settings limit the number of concurrent jobs per project on each node of the cluster.

To specify this job limit setting, in the Project Configuration Editor for the project, select the **Governing Rules: Default: Jobs** category, and specify the number of concurrent jobs that you want to allow for the project in each **Jobs per project** field. You can also specify a maximum number of interactive jobs (jobs executed by a direct user request) and scheduled jobs (jobs executed by a scheduled request). A value of -1 indicates that the number of jobs that can be executed has no limit.

Limiting the number of jobs per user session and per user account

If your users' job requests place a heavy burden on the system, you can limit the number of open jobs within Intelligence Server, including element requests, autoprompts, and reports for a user.

- To help control the number of jobs that can run in a project and thus reduce their impact on system resources, you can limit the number of concurrent jobs that a user can execute in a user session. For example, if the **Jobs per user session** limit is set to four and a user has one session open for the project, that user can only execute four jobs at a time. However, the user can bypass this limit by logging in to the project multiple times. (To prevent this, see the next setting, Jobs per user account limit.)

  To specify this setting, in the Project Configuration Editor for the project, select the **Governing Rules: Jobs** category, and type the number in the **Jobs per user session** field. A value of -1 indicates that the number of jobs that can be executed has no limit

- You can set a limit on the number of concurrent jobs that a user can execute for each project regardless of the number of user sessions that user has at the time. For example, if the user has two user sessions and the Jobs per user session limit is set to four, the user can run eight jobs. But if this **Jobs per user account** limit is set to five, that user can execute only five jobs, regardless of the number of times the user logs in to the system. Therefore, this limit can prevent users from circumventing the Jobs per user session limit by logging in multiple times.

© 2016, MicroStrategy Inc.

To specify this setting, in the Project Configuration Editor for the project, select the **Governing Rules: Jobs** category, and type the number of jobs per user account that you want to allow in the **Jobs per user account** field. A value of -1 indicates that the number of jobs that can be executed has no limit.

These two limits count the number of report, element, and autoprompt job requests that are executing or waiting to execute. Jobs that have finished, cached jobs, or jobs that returned in error are not counted toward these limits. If either limit is reached, any jobs the user submits do not execute and the user sees an error message.

Limiting the number of executing jobs per user and project

If your users tend to request jobs that do not place much burden on the system, you may want to limit only executing reports and data marts, and still allow users to answer autoprompts and issue element requests. You can limit the number of concurrent reports (both regular reports and dataset reports in a document) in a project per user account.

This limit is called **Executing jobs per user**. If the limit is reached for the project, new report requests are placed in the Intelligence Server queue until other jobs finish. They are then processed in the order in which they were placed in the queue, which is controlled by the priority map (see *Prioritizing jobs, page 340*).

To specify this limit setting, in the Project Configuration Editor for the project, select the **Governing Rules: Default: Jobs** category, and type the number of concurrent report jobs per user you want to allow in the **Executing jobs per user** field. A value of -1 indicates that the number of jobs that can be executed has no limit.

Limiting the maximum report execution time

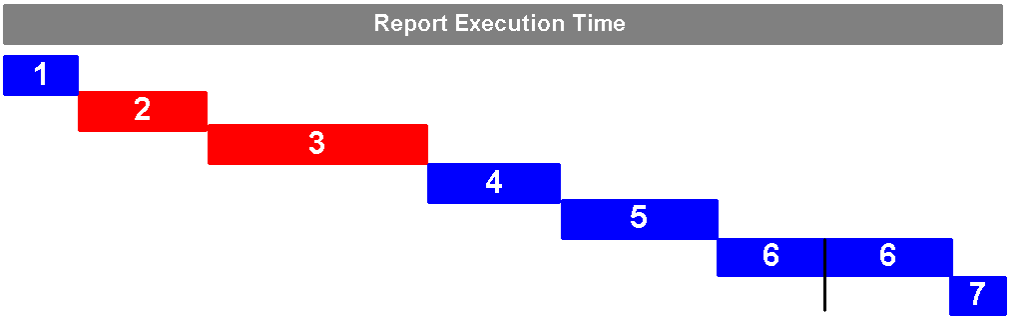
You can limit a job in Intelligence Server by specifying the maximum amount of time that a job can execute within a project. Intelligence Server cancels any jobs that exceed the limit.

To set this limit, in the Project Configuration Editor, select the **Governing Rules: Default: Result Sets** category, and specify the number of seconds in the **Intelligence Server Elapsed Time (sec)** fields. You can set different limits for ad-hoc reports and scheduled reports.

This limit applies to most operations that are entailed in a job from the time it is submitted to the time the results are returned to the user. If the job exceeds the limit, the user sees an error message and cannot view the report.

The figure below illustrates how job tasks make up the entire report execution time. In this instance, the time limit includes the time waiting for the user to complete report prompts. Each step is explained in the table below.

 © 2016, MicroStrategy Inc.



Step	Status	Comment
1	Waiting for Autoprompt	Resolving prompts
2*	Waiting (in queue)	Element request is waiting in job queue for execution
3*	Executing	Element request is executing on the database
4	Waiting for Autoprompt	Waiting for user to make prompt selections
5	Waiting (in queue)	Waiting in job queue for execution
6	Executing	Query engine executes SQL on database (can be multiple passes)
7	Executing	Analytical engine processes results

*Steps 2 and 3 are for an element request. They are executed as separate jobs. During steps 2 and 3, the original report job has the status “Waiting for Autoprompt.”

The following tasks are not shown in the example above because they consume very little time. However, they also count toward the report execution time.

- Element request SQL generation
- Report SQL generation
- Returning results from the database

For more information about the job processing steps, see *Processing jobs, page 32*.

Limiting a report's SQL per pass

You can limit a report’s SQL size per pass. This includes limits on the amount of time that each pass can take and the maximum size (in bytes) that the SQL statement can be. These limits are set in the VLDB properties, as described below. For more information about VLDB properties in general, see the *VLDB Properties* chapter in the Supplemental Reference for System Administration.

You can also limit the amount of memory that Intelligence Server uses during report SQL generation. This limit is set for all reports generated on the server. To set this limit, in the Project Configuration Editor, open the **Governing Rules: Default: Result Sets** category, and specify the **Memory consumption during SQL generation**. A value of -1 indicates no limit.

SQL Time Out (Per Pass) (database instance and report)

You can limit the amount of time that each pass of SQL can take within the data warehouse. If the time for a SQL pass reaches the maximum, Intelligence Server cancels the job and the user sees an error message. You can specify this setting at either the database instance level or at the report level.

To specify this setting, edit the VLDB properties for the database instance or for a report, expand **Governing settings**, then select the **SQL Time Out (Per Pass)** option. For details, see the *MicroStrategy Developer Help*.

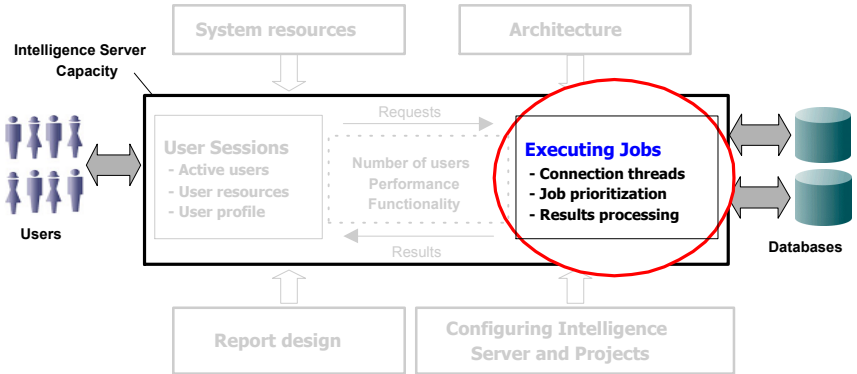
Maximum SQL Size (database instance)

You can limit the size (in bytes) of the SQL statement per pass before it is submitted to the data warehouse. If the size for a SQL pass reaches the maximum, Intelligence Server cancels the job and the user sees an error message. You can specify this setting at the database instance level.

To specify this, edit the VLDB properties for the database instance, expand **Governing settings**, then select the **Maximum SQL Size** option. For details, see the *MicroStrategy Developer Help*.

Managing job execution

The system's ability to execute jobs is limited by the available system resources and by how those resources are used by Intelligence Server.



 © 2016, MicroStrategy Inc.

This section discusses the different ways you have of managing job execution. These include:

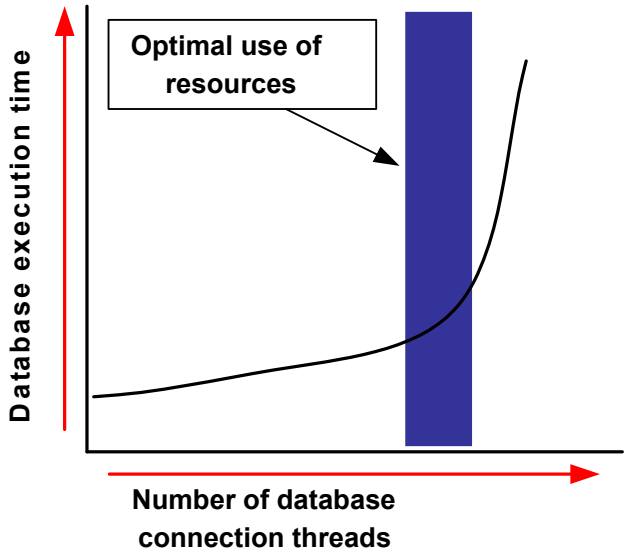
- *Managing database connection threads, page 337*
- *Prioritizing jobs, page 340*
- *Results processing, page 343* (the processing that Intelligence Server performs on results returned from the data warehouse)

Managing database connection threads

The main factor that determines job execution performance is the number of database connections that are made to the data warehouse. Report and element requests are submitted from Intelligence Server to the data warehouse through a database connection thread. Results of these requests are also returned to Intelligence Server through the database connection thread.

You must determine the number of threads that strikes a good balance between quickly serving each user request while not overloading the system. The overall goal is to prioritize jobs and provide enough threads so that jobs that must be processed immediately are processed immediately, and the remainder of jobs are processed as timely as possible. If your system has hundreds of concurrent users submitting requests, you must determine at what point to limit the number of database connection threads by placing user requests in a queue.

The number of available database connection threads falls in the range depicted as the Optimal use of resources in the illustration below.



To monitor whether the number of database connection threads in your system is effective, use the Database Connection Monitor. For more information about this tool, see *Monitoring database instance connections, page 9*. If all threads are “Busy” a high percentage of the time, consider increasing the number of connection threads as long as your data warehouse can handle the load and as long as Intelligence Server does not become overloaded.

Once you have the number of threads calculated, you can then set job priorities and control how many threads are dedicated to serving jobs meeting certain criteria.

Limiting and prioritizing the number of database connections

To set the number of database connection threads allowed at a time, modify the database instance used to connect to the data warehouse. Use the **Job Prioritization** tab in the Database Instance Editor and specify the number of high, medium, and low connections. The sum of these numbers is the total number of concurrent connection threads allowed between Intelligence Server and the data warehouse. These settings apply to all projects that use the selected database instance.

You should have at least one low-priority connection available, because low priority is the default job priority, and low-priority jobs can use only low-priority database connection threads. Medium-priority connection threads are reserved for medium- and high-priority jobs, and high-priority connection threads are reserved for high-priority jobs only. For more information about job priority, including instructions on how to set job priority, see *Prioritizing jobs, page 340*.

If you set all connections to zero, jobs are not submitted to the data warehouse. This may be a useful way for you to test whether scheduled reports are processed by Intelligence Server properly. Jobs wait in the queue and are not submitted to the data warehouse until you increase the connection number, at which point they are then submitted to the data warehouse. Once the testing is over, you can delete those jobs so they are never submitted to the data warehouse.

Optimizing database connection threads using ODBC settings

In addition to limiting the number of database connection threads created between Intelligence Server and the data warehouse, it is a good practice to efficiently use those connection threads once they are established. You want to ensure that the threads are being used and are not tied up by processes that are running too long. To optimize how those threads are used, you can limit the length of time they can be used by certain jobs. These limits are described below.

To set these limits, edit the database instance, then modify the database connection (at the bottom of the Database Instances dialog box), and on the Database Connections dialog box, select the **Advanced** tab. A value of 0 or -1 indicates no limit.

Maximum cancel attempt time

When a user runs a report that executes for a long time on the data warehouse, the user can cancel the job execution. This may be due to an error in the report's design, especially if it is in a project in a development environment, or the user may simply not want to wait any longer. If the cancel is not successful after 30 seconds, Intelligence Server deletes that job's database connection thread. The **Maximum cancel attempt time (sec)** field controls how long you want Intelligence Server to wait in addition to the 30 seconds before deleting the thread.

 © 2016, MicroStrategy Inc.

Maximum query execution time

This is the maximum amount of time that a single pass of SQL can execute on the data warehouse. When the SQL statement or fetch operation begins, a timer starts counting. If the **Maximum query execution time (sec)** limit is reached before the SQL operation is concluded, Intelligence Server cancels the operation.

This setting is very similar to the **SQL time out (per pass)** VLDB setting (see *SQL Time Out (Per Pass) (database instance and report), page 336*). That VLDB setting overrides the **Maximum query execution time (sec)** setting. This setting is made on the database connection and can be used to govern the maximum query execution time across all projects that use that connection. The VLDB setting can override this setting for a specific report.

Maximum connection attempt time

This is the maximum amount of time that Intelligence Server waits while attempting to connect to the data warehouse. When the connection is initiated, a timer starts counting. If the **Maximum connection attempt time (sec)** limit is reached before the connection is successful, the connection is canceled and an error message is displayed.

Limiting database connection caches

Establishing a database connection thread is expensive in terms of time and resources. Because of this, Intelligence Server caches the threads so that every SQL pass and job execution it performs does not need to create a new connection. Rather, those processes simply use an existing cached thread. However, the RDBMS may, after a certain time limit, delete the connection threads without notifying Intelligence Server. If this happens and an Intelligence Server job tries to use a cached connection thread, the user sees an error message. To avoid this, you can limit the length of time that a database connection cache can exist. You can limit the maximum lifetime of a database connection (see *Connection lifetime, page 339*), and you can limit the amount of time an inactive database connection remains open (see *Connection idle timeout, page 340*).

To set these limits, edit the database instance, then modify the database connection (at the bottom of the Database Instances dialog box), and on the Database Connections dialog box, select the **Advanced** tab. For these settings, a value of -1 indicates no limit, and a value of 0 indicates that the connection is not cached and is deleted immediately when execution is complete.

Connection lifetime

The **Connection lifetime (sec)** limit is the maximum amount of time that a database connection thread remains cached. The **Connection lifetime** should be shorter than the data warehouse RDBMS connection time limit. Otherwise the RDBMS may delete the connection in the middle of a job.

When the **Connection lifetime** is reached, one of the following occurs:

- If the database connection has a status of **Cached** (it is idle, but available) when the limit is reached, the connection is deleted.

© 2016, MicroStrategy Inc.

- If the database connection has a status of **Busy** (it is executing a job) when the limit is reached, the connection is deleted as soon as the job completes. The database connection does not go into a Cached state.

Connection idle timeout

The **Connection idle timeout (sec)** limit is the amount of time that an inactive connection thread remains cached in Intelligence Server until it is terminated. When a database connection finishes a job and no job is waiting to use it, the connection becomes cached. If the connection remains cached for longer than this timeout limit, the database connection thread is then deleted. This prevents connections from tying up data warehouse and Intelligence Server resources if they are not needed.

Prioritizing jobs

Job priority defines the order in which jobs are processed. Jobs are usually executed as first-come, first-served. However, your system probably has certain jobs that need to be processed before other jobs.

Job priority does not affect the amount of resources a job gets once it is submitted to the data warehouse. Rather, it determines whether certain jobs are submitted to the data warehouse before other jobs in the queue.

For example, an executive in your company runs reports at unplanned times and you want to ensure that these reports are immediately processed. If no priority is set for the executive's reports, they are processed with the other jobs in the system. Depending on data warehouse activity, this may require some wait time. If you assign a high priority to all jobs from the executive's user group, Intelligence Server processes and submits those jobs to the data warehouse first, rather than waiting for other jobs to finish.

Intelligence Server processes a job on a database connection that corresponds to the job's priority. If no priority is specified for a job, Intelligence Server processes the job on a low-priority connection. For example, jobs with high priority are processed by high-priority connections, and jobs with low or no priority are processed by a low-priority connection. For information about setting database connection thread priority, see *Limiting and prioritizing the number of database connections, page 338*.

Intelligence Server also engages in connection borrowing when processing jobs. *Connection borrowing* occurs when Intelligence Server executes a job on a lower priority connection because no connections that correspond to the job's priority are available at execution time. High-priority jobs can run on high-, medium-, and low-priority connections. Likewise, medium-priority jobs can run on medium- and low-priority connections.

When a job is submitted and no connections are available to process it, either with the same priority or with a lower priority, Intelligence Server places the job in queue and then processes it when a connection becomes available.

You can set jobs to be high, medium, or low priority, by one or more of the following variables:

 © 2016, MicroStrategy Inc.

- **Request type**: Report requests and element requests can have different priority (*Prioritizing jobs by request type, page 341*).
- **Application type**: Jobs submitted from different MicroStrategy applications, such as Developer, Scheduler, MicroStrategy Web, or Narrowcast Server, are processed according to the priority that you specify (*Prioritizing jobs by MicroStrategy application type, page 342*).
- **User group**: Jobs submitted by users in the groups you select are processed according to the priority that you specify (*Prioritizing jobs by user group, page 342*).
- **Cost**: Jobs with a higher resource cost are processed according to the priority that you specify (*Prioritizing jobs by report cost, page 342*). Job cost is an arbitrary value you can assign to a report that represents the resources used to process that job.
- **Project**: Jobs submitted from different projects are processed according to the priority that you specify (*Prioritizing jobs by project, page 343*).

These variables allow you to create sophisticated rules for which job requests are processed first. For example, you could specify that any element requests are high priority, any requests from your test project are low priority, and any requests from users in the Developers group are medium priority.

A job is processed at the highest priority assigned to it by any rules. For example, if you set all jobs from your test project at low priority, and all jobs from users in the Developers group at medium priority, jobs in the test project that are requested by users in the Developers group are processed at medium priority.

To set job prioritization rules

1. On the Intelligence Server machine, in Developer, log in to a project source. You must log in as a user with administrative privileges.
2. Expand the **Administration** folder, then expand **Configuration Managers**, and then select **Database Instances**.
3. Right-click the database instance used to connect to the data warehouse and select **Prioritization**. The Database Instances Editor opens with the Job Prioritization tab selected. Any prioritization rules that have already been created are listed in the tab.
4. To add new job prioritization rules, click **New**. The Job Prioritization Wizard opens. For detailed instructions on using the wizard, press **F1** to view the Help for each page of the wizard.

Prioritizing jobs by request type

You can select whether element requests or report requests are processed first. For example, you may want element requests to be submitted to the data warehouse before report requests, because element requests are generally used in prompts and you do not want users to have to wait long while prompt values load. In this case you might specify all element requests to be processed at a high priority by default, and all report requests to be processed at a low priority by default.

© 2016, MicroStrategy Inc.

Prioritizing jobs by MicroStrategy application type

You can assign a different priority to jobs submitted from Developer, MicroStrategy Web or Web Universal, Scheduler, and Narrowcast Server. All jobs submitted from the specified application use the specified priority. For example, you may want report designers to be able to quickly test their reports, so you may specify that all jobs that are submitted from Developer are processed at a high priority.

Prioritizing jobs by user group

You can assign a different priority to jobs submitted from different MicroStrategy user groups. For example, you can assign all jobs from users in the Executive user group to be processed at a high priority.

Prioritizing jobs by report cost

Report cost is an arbitrary value that you can assign to a report to help determine its priority in relation to other requests. If you choose to use report cost as a priority variable, you must define a set of priority groups based on report cost. The default priority groups are:

- Light: reports with costs between 0 and 334
- Medium: reports with costs between 335 and 666
- Heavy: reports with costs between 667 and 999

The set of cost groupings must cover all values from 0 to 999. You can then assign a priority level to each priority group. For example, you can set heavy reports to low priority, because they are likely to take a long time to process, and set light reports to high priority, because they do not place much strain on the system resources.

Once you determine the cost groupings, you can set the report cost value on individual reports. For example, you notice that a report requires significantly more processing time than most other reports. You can assign it a report cost of 900 (heavy). In this sample configuration, the report has a low priority. For factors that may help you determine the cost of a report, see *Results processing, page 343*.

You set the cost of a report in the report's Properties dialog box, in the **Priority** category. You must have system administrator privileges to set the cost of a report.

To set the cost for a report

1. In Developer, right-click the report and select **Properties**. The Properties dialog box opens.
2. Select the **Priority** category.
3. In the **Report Cost** field, type the cost of the report. Higher numbers indicate a report that uses a great deal of system resources. Lower numbers indicate a less resource-intensive report.

 © 2016, MicroStrategy Inc.

4 Click **OK**. The Properties dialog box closes and the cost of the report is set.

Prioritizing jobs by project

You can assign a different priority to reports from different projects. For example, you may want all jobs submitted from your production project to have a medium priority, so that they take precedence over reports from your test project.

Results processing

When Intelligence Server processes results that are returned from the data warehouse, several factors determine how much of the machine's resources are used. These factors include:

- Whether Intelligence Server is using thread balancing (see *Intelligence Server thread balancing, page 343*)
- The size of the report (see *Limiting the maximum report size, page 343*)
- Whether the report is an Intelligent Cube (see *Limiting the size and number of Intelligent Cubes, page 346*)
- Whether the report is imported from an external data source (see *Limiting the memory used during data fetching, page 347*)

Intelligence Server thread balancing

By default, threads within Intelligence Server process tasks in the order that they are received. You can configure Intelligence Server to allocate threads to processes, such as object serving, element serving, SQL generation, and so forth, that need them most, while less loaded processes can return threads to the available pool.

To enable thread balancing for Intelligence Server, in the Intelligence Server Configuration Editor, in the **Server Definition: Advanced** category, select the **Balance MicroStrategy Server threads** check box.

Limiting the maximum report size

A report instance is the version of the report results that Intelligence Server holds in memory for cache and working set results. The size of the report instance is proportional to the size of the report results, that is, the row size multiplied by the number of rows.

The row size depends on the data types of the attributes and metrics on the report. Dates are the largest data type. Text strings, such as descriptions and names, are next in size, unless the description is unusually long, in which case they may be larger than dates. Numbers, such as IDs, totals, and metric values, are the smallest.

The easiest way to estimate the amount of memory that a report uses is to view the size of the cache files using the Cache Monitor in Developer. The Cache Monitor shows the size of the report results in binary format, which from testing has proven to be 30 to 50 percent of the actual size of the report instance in memory. For instructions on how to

© 2016, MicroStrategy Inc.

use the Cache Monitor to view the size of a cache, see *Monitoring result caches, page 409*.

Intelligence Server allows you to govern the size of a report or request in the following ways:

- *Limiting the number of report result rows, page 344*
- *Limiting the number of element rows, page 345*
- *Limiting the number of intermediate rows, page 346*

Like all requests, large report instances are also governed by the Memory Contract Manager settings. For more information about Memory Contract Manager, see *Governing Intelligence Server memory use with Memory Contract Manager, page 316*.

Limiting the number of report result rows

Reports with a large number of result rows can take up a great deal of memory at run time. For example, your data warehouse may contain daily sales data for thousands of items over several years. If a user attempts to build a report that lists the revenue from every item for every day in the data warehouse, the report may use all available Intelligence Server memory.

You can limit a report's size in Intelligence Server by setting a maximum limit on the number of rows that a report can contain. This setting is applied by the Query Engine when retrieving the results from the database. If the report exceeds this limit, the report is not executed and an error message is displayed.

To set the maximum number of result rows for all reports, data marts, and Intelligent Cubes in a project, in the Project Configuration Editor, expand the **Governing Rules: Default: Result Sets** category, and type the maximum number in the appropriate **Final Result Rows** field. You can set different limits for standard reports, Intelligent Cubes, and data marts.

You can also set the result row limit for a specific report in that report's VLDB properties. The VLDB properties limit for a report overrides the project limit. For example, if you set the project limit at 10,000 rows, but set the limit to 20,000 rows for a specific report that usually returns more than 10,000 rows, users are able to see that report without any errors.

To set the result set limit for a specific report

1. In Developer, right-click the report to set the limit for and select **Edit**. The Report Editor opens.
2. From the **Data** menu, select **VLDB properties**. The VLDB Properties dialog box opens.
3. Expand the **Governing** settings, then select **Results Set Row Limit**.

 © 2016, MicroStrategy Inc.

4 Make sure the **Use default inherited value** check box is cleared.

5 In the **Results Set Row Limit** field, type the limit.

6 Click **Save and Close** to save the VLDB properties and close the VLDB Properties dialog box.

7 Click **Save and Close** to save the report and its changed VLDB properties.

Limiting the number of element rows

Another way that you can limit the size of a request is to limit the number of element rows returned at a time. Element rows are returned when a user accesses a report prompt, and when using the Data Explorer feature in Developer.

Element rows are incrementally fetched, that is, returned in small batches, from the data warehouse to Intelligence Server. The size of the increment depends on the maximum number of element rows specified in the client. Intelligence Server incrementally fetches four times the number for each element request.

For more information about element requests, such as how they are created, how incremental fetch works, and the caches that store the results, see *Element caches, page 440*.

MicroStrategy recommends that you set the element row limit to be larger than the maximum number of attribute element rows that you expect users to browse. For example, if the Product table in the data warehouse has 10,000 rows that users want to browse and the Order table has 200,000 rows that you do not expect users to browse, you should set this limit to 11,000. Intelligence Server incrementally fetches the element rows. If the element rows limit is reached, the user sees an error message and cannot view the prompt or the data.

To set the maximum number of element rows returned for all element requests in a project in Developer, in the Project Configuration Editor for that project, expand the **Governing Rules: Default: Result Sets** category and type the number in the **All element browsing result rows** field.

To set the number of objects returned for requests in MicroStrategy Web

1 In MicroStrategy Web, log in to a project as a user with the Web Administration privilege.

2 Click the **MicroStrategy** icon, then select **Preferences**. The Web Preferences page opens.

3 Select **Project defaults**, and then select the **General** category.

4 In the Incremental Fetch section, specify the values in the **Maximum number of attribute elements per block** and **Maximum number of report objects per block** fields.

5 Click **OK**. The new maximum values are saved.

© 2016, MicroStrategy Inc.

Limiting the number of intermediate rows

You can limit a report's size on Intelligence Server by setting a maximum number of intermediate result rows that are allowed in Intelligence Server. This limit does not apply to the rows in intermediate or temporary tables created in the data warehouse. Rather, it controls the number of rows held in memory in the Analytical Engine processing unit of Intelligence Server for analytic calculations that cannot be done on the database. Lowering this setting reduces the amount of memory consumed for large reports. If the limit is reached, the user sees an error message and cannot view the report. For example, this may happen when you add a complex subtotal to a large report or when you pivot a large report.

To specify this limit for all reports in a project, in the Project Configuration Editor, select the **Governing Rules: Default: Result Sets** category and type the number in the **All intermediate result rows** box.

You can also set the intermediate row limit for a specific report in that report's VLDB properties. The VLDB properties limit for the report overrides the project limit. For example, if you set the project limit at 10,000 rows but set the limit to 20,000 rows for a specific report that usually returns more than 10,000 rows, users are able to see that report without any errors.

To set the Intermediate row limit for a specific report

1. In Developer, right-click the report to set the limit for and select **Edit**. The Report Editor opens.
2. From the **Data** menu, select **VLDB properties**. The VLDB Properties dialog box opens.
3. Expand the **Governing** settings, then select **Intermediate Row Limit**.
4. Make sure the **Use default inherited value** check box is cleared.
5. In the **Intermediate Row Limit** field, type the limit.
6. Click **Save and Close** to save the VLDB properties and close the VLDB Properties dialog box.
7. Click **Save and Close** to save and close the report.

Limiting the size and number of Intelligent Cubes

If you have purchased OLAP Services licenses from MicroStrategy, your report designers can create Intelligent Cube reports. These Intelligent Cubes must be stored in Intelligence Server memory for reports to access their data. This may cause a shortage of memory for other processes on the Intelligence Server machine.

You can govern the amount of resources used by Intelligent Cubes by limiting the amount of memory used by Intelligent Cubes and by limiting the number of Intelligent Cubes that can be loaded into memory.

 © 2016, MicroStrategy Inc.

To specify these settings, in the Project Configuration Editor for the project, select the **Cubes: General** category and type the new values in the **Maximum RAM usage (MBytes)** and **Maximum number of cubes** fields. For detailed information on governing Intelligent Cube memory usage, see *Defining memory limits for Intelligent Cubes, page 469*.

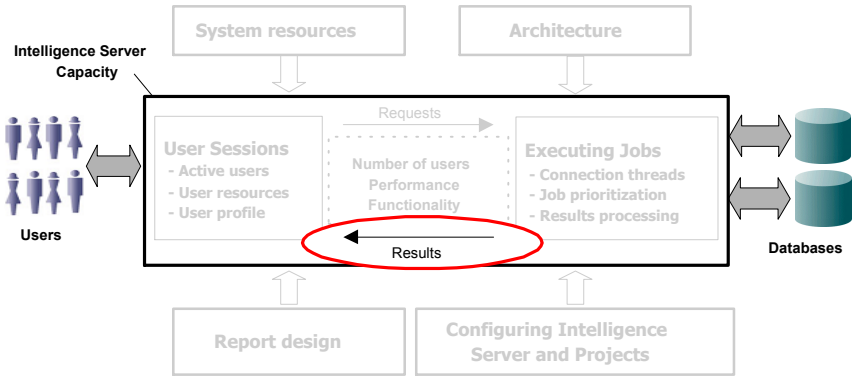
Limiting the memory used during data fetching

Certain MicroStrategy features enable you to fetch data from external data sources, such as web services, MDX cubes, or Excel spreadsheets. When data is fetched from one of these data sources, it is temporarily stored in Intelligence Server memory while being converted to a report. This can cause a shortage of memory for other processes on the Intelligence Server machine.

You can govern the amount of memory used for an individual data fetch in the Project Configuration Editor. Select the **Governing Rules: Default: Result Sets** category, and type the new value in the **Memory consumption during data fetching (MB)** field. The default value is -1, indicating no limit.

Governing results delivery

After Intelligence Server processes the results of a job (see *Managing job execution, page 336*), it then delivers the results to the user. In a three-tier system, results delivery uses very little of the system resources. Most of the tuning options for results delivery are focused on a four-tier system involving MicroStrategy Web.



To deliver results, when a report is first run or when it is manipulated, Intelligence Server generates XML and sends it to the MicroStrategy Web server. The Web server then translates the XML into HTML for display in the user's web browser.

You can set limits in two areas to control how much information is sent at a time. The lower of these two settings determines the maximum size of results that Intelligence Server delivers at a time:

- How many rows and columns can be displayed simultaneously in MicroStrategy Web (see *Limiting the information displayed at one time, page 348*)

© 2016, MicroStrategy Inc.

- How many XML cells in a result set can be delivered simultaneously (see *Limiting the number of XML cells, page 349*)

The following settings also govern results delivery:

- The maximum size of a report that can be exported (see *Limiting export sizes, page 350* and *Limiting the memory consumption for file generation, page 350*)
- The number of XML drill paths in a report (see *Limiting the total number of XML drill paths, page 351*)

Like all requests, displayed and exported reports are also governed by the Memory Contract Manager settings. For more information about Memory Contract Manager, see *Governing Intelligence Server memory use with Memory Contract Manager, page 316*.

Limiting the information displayed at one time

In MicroStrategy Web, if a report contains a large amount of data, it can use a great deal of the system resources and take a significant amount of time before it is displayed to the user. You can lessen the impact of these large reports by limiting the maximum number of rows and columns that are displayed. If a report's result set is larger than these limits, the report is broken into pages (increments) that are fetched from the server one at a time.

The size of these increments can be set as project defaults by the MicroStrategy Web administrator. Users with the Web Change User Preferences privilege can also customize these sizes. For more information about these increments, see the information on Incremental fetch in the MicroStrategy Web Help.

To limit the number of rows and columns for all users

1. In MicroStrategy Web, log in to a project as a user with the Web Administration privilege.
2. Click the **MicroStrategy** icon, then click **Preferences**. The Web Preferences page opens.
3. Select **Project defaults**, and then select the **Grid display** category.
4. Specify the values in the **Maximum rows in grid** and **Maximum columns in grid** fields.
5. Click **OK**. The new maximum values are saved.

 © 2016, MicroStrategy Inc.

To limit the number of rows and columns for one user

1. In MicroStrategy Web, log in to a project as a user with the Web Change User Preferences privilege.
2. Click the **MicroStrategy** icon, then click **Preferences**. The Web Preferences page opens.
3. Select the **Grid display** category.
4. Specify the values in the **Maximum rows in grid** and **Maximum columns in grid** fields.

   If the user sets the number of rows and columns too high, the number of XML cells limit that is set in Intelligence Server (see *Limiting the number of XML cells, page 349*) governs the size of the result set.

5. Click **OK**. The new maximum values are saved for that user.

Limiting the number of XML cells

When large report result sets are generated into XML, they can require a significant amount of Intelligence Server memory. MicroStrategy Web handles this by implementing the incremental fetch feature (see *Limiting the information displayed at one time, page 348*). You can also govern the result set's size by setting the **Maximum number of XML cells** at the Intelligence Server level**.** This determines the maximum number of cells that can be returned from Intelligence Server to the Web server at a time. For this limit, the number of cells is the number of rows multiplied by the number of metric columns. Attribute cells are not considered.

For example, if the XML limit is set at 10,000 and a report has 100,000 metric cells, the report is split into 10 pages. The user clicks the page number to view the corresponding page.

Additionally, when users export large reports from MicroStrategy Web as formatted data, the XML is generated in batches. This XML limit determines how large the batches are. Depending on this XML limit, Intelligence Server behaves differently:

- If the limit is smaller, it takes a longer time to generate the XML because it is generated in small batches, which use less memory and system resources.
- If the limit is larger, it takes a shorter time to generate the XML because it is generated in fewer, but larger, batches, which use more memory and system resources.

To set the XML limit, in the Intelligence Server Configuration Editor, select the **Governing Rules: Default: File Generation** category, then specify the **Maximum number of XML cells**. You must restart Intelligence Server for the new limit to take effect.

© 2016, MicroStrategy Inc.

Limiting export sizes

When users export a report from MicroStrategy Web, the results are not constrained by the incremental fetch limit or the XML limit. To govern the size of reports that can be exported, you can set limits on the number of cells for various export formats.

To limit the number of rows and columns for all users

1. In MicroStrategy Web, log in to a project as a user with the Web Administration privilege.
2. Click the **MicroStrategy** icon, then click **Preferences**. The Web Preferences page opens.
3. Select **Project defaults**, and then select the **Export Reports** category.
4. Specify the values in the **Maximum number of cells to export to plain text** and **Maximum number of cells to export to HTML and Excel with formatting** fields.
5. Click **OK**. The new maximum values are saved.

Limiting the memory consumption for file generation

Exporting a report to a different format can consume a great deal of memory. The amount of memory available for use by exporting files from MicroStrategy Web is governed by the maximum memory consumption limits in the Intelligence Server Configuration Editor. If an export attempts to use more memory than these settings allow, the export fails with the error message "MicroStrategy Intelligence Server cannot handle your request because a memory request has exceeded the configured limit. Please contact the server administrator."

The more formatting an exported report has, the more memory it consumes. When exporting large reports the best options are plain text or CSV file formats because formatting information is not included with the report data. In contrast, exporting reports as Excel with formatting uses a significant amount of memory because the exported Excel file contains both the report data and all the formatting data. For more information about exporting reports, see *What happens when I export a report from MicroStrategy Web?, page 44*.

Because Excel export uses significantly more memory than other export formats, you can limit the size of reports exported to Excel from Developer as well as from Web. The default memory consumption limit is 100 MB.

To set the maximum memory consumption limits for exporting reports from Web, in the Intelligence Server Configuration Editor, select the **Governing Rules: Default: File Generation** category, and specify the **Maximum memory consumption** for the XML, PDF, Excel, and HTML files.

 © 2016, MicroStrategy Inc.

Depending on your Memory Contract Manager settings, an export can use less memory than specified by these settings and still be denied because of a lack of memory. For more information about Memory Contract Manager, see *Governing Intelligence Server memory use with Memory Contract Manager, page 316*.

To set the maximum memory consumption for Excel file generation

1. In Developer, log in to a project source using an account with the Configure Server Basic privilege.
2. From the **Tools** menu, select **Project Source Manager**. The Project Source Manager opens.
3. Select the project source and click **Modify**. The Project Source Manager for that project source opens.
4. On the **Governing** tab, in the **Export to Excel** section, select **Use custom value**. In the **Maximum RAM Usage (MB)** field, specify the maximum memory consumption.
5. Click **OK**. The Project Source Manager closes and the limit is saved.

Limiting the total number of XML drill paths

Another way that you can prevent reports from consuming too much memory is to limit the number of XML drill paths allowed on reports in MicroStrategy Web products. The default drill map for reports uses all attributes included in hierarchies marked as drill hierarchies. Report designers can significantly reduce the size of an attribute's drill path by modifying a report's drill map to include fewer drill options. You can also impose a limit for all reports coming from MicroStrategy Web products by setting the Maximum number of XML drill paths.

For more information about customizing drill maps, see the Advanced Reporting Guide.

To set this limit, in the Intelligence Server Configuration Editor, select the **Governing Rules: Default: File Generation** category, then specify the **Maximum number of XML drill paths**. You must restart Intelligence Server for the new limit to take effect.

Disabling XML caching for a project may have a negative effect on performance, especially for large reports. For more information, see *XML caches, page 404* and *ACLs and personalized drill paths in MicroStrategy Web, page 66*.

Tuning your system for in-memory datasets

You can import large datasets into your Intelligence Server's memory as Intelligent Cubes, and divide the Intelligent Cubes into multiple segments. These segments, called partitions, are processed simultaneously, distributed across the processor cores of your Intelligence Server.

By storing your data in your Intelligence Server's memory and processing the data using all the server's processor cores, you can analyze large and complex datasets with very fast response times.

The following sections cover the settings you can configure to improve the performance of your in-memory datasets:

- *Configuring Intelligence Server for in-memory datasets, page 352*
- *Configuring your projects for in-memory datasets, page 352*

Configuring Intelligence Server for in-memory datasets

To ensure the best performance for your partitioned in-memory datasets, you can configure the following settings for your Intelligence Server:

- Consider increasing the number of database connections that Intelligence Server uses to connect to data sources. When users import data into Intelligence Server's memory, the job to connect to the data source is given a low priority. To allow Intelligence Server to retrieve large datasets, you can increase the number of low-priority database connections that Intelligence Server can make.

  For background information on prioritizing jobs, see *Prioritizing jobs*. For background information on changing the number of database connections, see *Limiting and prioritizing the number of database connections*.

- Consider increasing the maximum time that a database query is allowed to run, to ensure that the Intelligence Server has more time to retrieve large datasets from the data source. For background information on increasing the execution time for database queries, see *Optimizing database connection threads using ODBC settings*

Configuring your projects for in-memory datasets

For each of your projects that uses in-memory datasets, you can make the following changes to improve the performance of your in-memory datasets:

- Increase the maximum size of the datasets that your users can import. If your users need to import large datasets into a project, you can increase the limit on the size of the dataset that they can import. For steps to increase this limit, see *Defining limits for Intelligent Cubes created using the Import Data feature*.

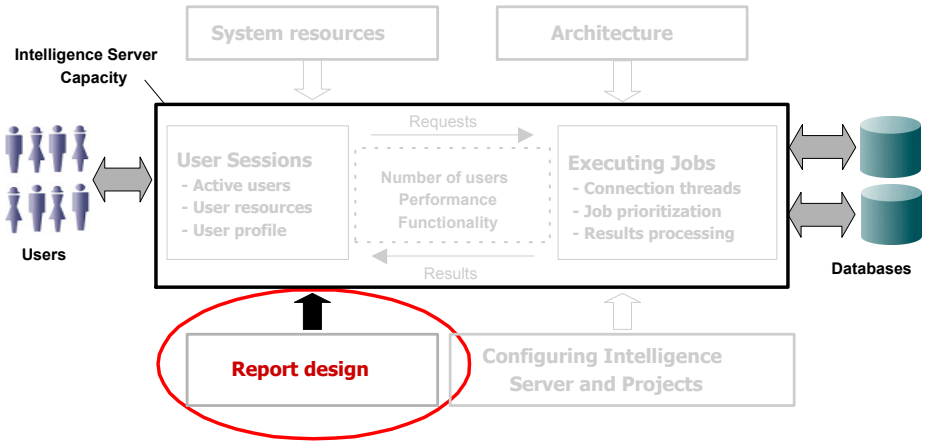
- Enable parallel queries for the reports in your project, so that Intelligence Server can execute database queries in parallel and retrieve more data from your database. For

 © 2016, MicroStrategy Inc.

steps to enable parallel queries, and to define the maximum number of parallel queries that can be run for every report, see the Supplemental Reference for System Administration.

Designing reports

In addition to the fact that large reports can exert a heavy toll on system performance, a report's design can also affect it. Some features consume more of the system's capacity than others when they are used.



Some report design features that can use a great deal of system resources include:

- Complex analytic calculations (*Analytic complexity, page 353*)
- Subtotals (*Subtotals, page 354*)
- Page-by (*Page-by feature, page 354*)
- Prompt complexity (*Prompt complexity, page 354*)
- Report Services documents (*Report Services documents, page 354*)
- Intelligent Cubes (*Intelligent Cubes, page 355*)

Analytic complexity

Calculations that cannot be done with SQL in the data warehouse are performed by the Analytical Engine in Intelligence Server. These may result in significant memory use during report execution. Some analytic calculations (such as `AvgDev`) require the entire column of the fact table as input to the calculation. The amount of memory used depends on the type of calculation and the size of the report that is used. Make sure your report designers are aware of the potential effects of these calculations.

Subtotals

The amount of memory required to calculate and store subtotals can be significant. In some cases, the size of the subtotals can surpass the size of the report result itself.

The size of the subtotals depends on the subtotaling option chosen, along with the order and the number of unique attributes. The easiest way to determine the number of subtotals being calculated is to examine the number of result rows added with the different options selected in the Advanced Subtotals Options dialog box. To access this dialog box, view the report in Developer, then point to **Data**, then **Subtotals**, and then choose **Advanced**. For more detailed information about the different subtotal options, see the *Reports* chapter in the Advanced Reporting Guide.

Subtotals can use a great deal of memory if you select the **All Subtotals** option in the **Pages** drop-down list. This option calculates all possible subtotal calculations at runtime and stores the results in the report instance. MicroStrategy recommends that you encourage users and report designers to use less taxing options for calculating subtotals across pages, such as **Selected Subtotals** and **Grand Total**.

Page-by feature

If designers or users create reports that use the page-by feature, they may use significant system resources. This is because the entire report is held in memory even though the user is seeing only a portion of it at a time. To lessen the potential effect of using page-by with large reports, consider splitting those reports into multiple reports and eliminating the use of page-by. For more information about page-by, see the Advanced Reporting Guide.

Prompt complexity

Each attribute element or hierarchy prompt requires an element request to be executed by Intelligence Server. The number of prompts used and the number of elements returned from the prompts determine how much load is placed on Intelligence Server. Report designers should take this into account when designing prompted reports.

In addition to limiting the number of elements returned from element requests (as described in *Limiting the number of element rows, page 345*), you should make sure your element caches are being used effectively. For information on managing element caches, including instructions, see *Element caches, page 440*.

Report Services documents

Report Services documents may contain multiple reports. Executing a document can result in several report requests being submitted simultaneously. In addition, Flash dashboards can often require a great deal of resources to render.

To limit the effect of Report Services documents on the system, consider enabling document caching. If the documents are cached on Intelligence Server, less load is

 © 2016, MicroStrategy Inc.

placed on the data warehouse and on the Intelligence Server machine. For information about document caching, including instructions, see *Result caches, page 400*.

Intelligent Cubes

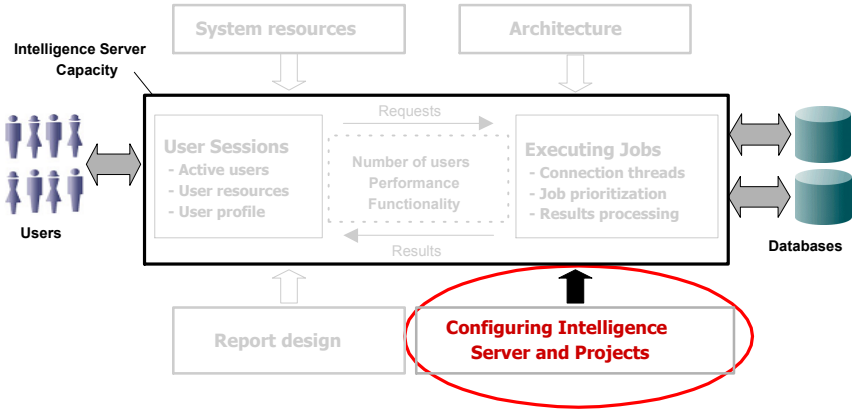
With OLAP Services features, your report designers can create Intelligent Cube reports. These reports allow data to be returned from the data warehouse, stored in Intelligence Server memory, and then shared among multiple reports.

Because Intelligent Cubes must be loaded into memory to be used in reports, they can use a great deal of system resources. Make sure your report designers are familiar with the Intelligent Cube design best practices found in *Best practices for reducing Intelligent Cube memory size, page 466*.

You can also restrict the number and size of Intelligent Cubes that can be loaded at once. For instructions, see *Limiting the size and number of Intelligent Cubes, page 346*.

Configuring Intelligence Server and projects

At times you may need to adjust settings in the MicroStrategy system, either as a result of changes to the system or to improve an aspect of system efficiency. This section provides an overview of the governing settings throughout the system.



These governors are arranged by where in the interface you can find them.

- Intelligence Server Configuration Editor (including project distribution settings) (*Intelligence Server Configuration Editor, page 356*)
- Project Configuration Editor (*Project Configuration Editor, page 361*)
- Database connections (*Database connection, page 365*)
- VLDB settings (*VLDB settings, page 366*)

© 2016, MicroStrategy Inc.

Intelligence Server Configuration Editor

To set the following governors in the Intelligence Server Configuration Editor, right-click the project source, select **Configure MicroStrategy Intelligence Server**, then select the category as described below.

Only the categories and settings in the Intelligence Server Configuration Editor that affect system scalability are described below. Other categories and settings that appear in the Intelligence Server Configuration Editor are described elsewhere in this guide, and in the Help for the editor. (From within the editor, press **Help**.)

Server definition: General category in Intelligence Server configuration

Governor	Description	See page
Number of network threads	Controls the number of network connections available for communication between Intelligence Server and the client, such as Developer or MicroStrategy Web.	*How the network can affect performance*

Server definition: Advanced category in Intelligence Server configuration

Governor	Description	See page
Backup frequency (minutes)	Controls the frequency (in minutes) at which cache and History List messages are backed up to disk. A value of 0 means that cache and history messages are backed up immediately after they are created.	*Cache duration (Hours)*
Balance MicroStrategy Server threads	Controls whether threads in Intelligence Server are allocated to processes such as object serving, element serving, SQL generation, and so on that need them most, while processes with lighter loads can return threads to the available pool.	*Results processing*
Cache lookup cleanup frequency (sec)	Cleans up the cache lookup table at the specified frequency (in seconds). This reduces the amount of memory the cache lookup table consumes and the time Intelligence Server takes to back up the lookup table to disk.	*Cache duration (Hours)*
Project failover latency (min.)	The amount of time (the delay) before the project is loaded on another server to maintain minimum level availability.	*Project failover latency*
Configuration recovery latency (min.)	When the conditions that caused a project failover disappear, the failover configuration reverts automatically to the original configuration. This setting is the amount of time (the delay) before the failover configuration reverts to the original configuration.	*Configuration recovery latency*
Enable performance monitoring	Configures additional MicroStrategy-specific monitors in Windows Performance Monitor.	*Monitoring memory use with Performance Monitor*

Governing Rules: Default: General category in Intelligence Server configuration

 © 2016, MicroStrategy Inc.

Governor	Description	See page
Maximum number of jobs	The maximum concurrent number of jobs that can exist on an Intelligence Server.	*Limiting the total number of jobs*
Maximum number of interactive jobs	Limits the number of concurrent interactive (nonscheduled) jobs that can exist on this Intelligence Server. A value of -1 indicates no limit.	*Limiting the total number of jobs*
Maximum number of scheduled jobs	Limits the number of concurrent scheduled jobs that can exist on this Intelligence Server. A value of -1 indicates no limit.	*Limiting the total number of jobs*
Maximum number of user sessions	The maximum number of user sessions (connections) for an Intelligence Server. A single user account may establish multiple sessions to an Intelligence Server.	*Limiting the number of user sessions on Intelligence Server*
User session idle time (sec)	The time allowed for a Developer user to remain idle before his or her session is ended. A user session is considered idle when it submits no requests to Intelligence Server.	*Limiting user session idle times*
Web user session idle time (sec)	The time allowed for a Web user to remain idle before his or her session is ended. **Note**: If designers will be building Report Services documents and dashboards in MicroStrategy Web, set the **Web user session idle time (sec)** to 3600 to avoid a project source timeout.	*Limiting user session idle times*
For Intelligence Server and history list governing, exclude reports embedded in Report Services documents from the counts	Do not include reports submitted as part of a document in the count of jobs for the job limits.	*Excluding document datasets from the job limits*
Background Execution: Enable background execution of documents after their caches are hit.	If selected, when a document cache is hit, Intelligence Server displays the cached document and re-executes the document in the background. If this option is cleared, when a document cache is hit, Intelligence Server displays the cached document and does not re-execute the document until a manipulation is performed. By default this option is cleared.	

Governing Rules: Default: File Generation category in Intelligence Server configuration

Governor	Description	See page
XML Generation: Maximum number of XML cells	The maximum number of XML cells in a report result set that Intelligence Server can send to the MicroStrategy Web products at a time. When this limit is reached, the user sees an error message along with the partial result set. The user can incrementally fetch the remaining cells.	*Limiting the number of XML cells*

Governor	Description	See page
XML Generation: Maximum number of XML drill paths	The maximum number of attribute elements that users can see in the drill across menu in MicroStrategy Web products. If this setting is set too low, the user does not see all the available drill attributes.	*Limiting the total number of XML drill paths*
XML Generation: Maximum memory consumption for XML (MB)	The maximum amount of memory (in megabytes) that Intelligence Server can use to generate a report or document in XML. If this limit is reached, the XML document is not generated and the user sees an error message.	*Limiting the memory consumption for file generation*
PDF Generation: Maximum memory consumption for PDF files (MB)	The maximum amount of memory (in megabytes) that Intelligence Server can use to generate a report or document in PDF. If this limit is reached, the PDF document is not generated and the user sees an error message.	*Limiting the memory consumption for file generation*
Excel Generation: Maximum memory consumption for Excel files (MB)	The maximum amount of memory (in megabytes) that Intelligence Server can use to generate a report or document in Excel. If this limit is reached, the Excel document is not generated and the user sees an error message.	*Limiting the memory consumption for file generation*
HTML Generation: Maximum memory consumption for HTML files (MB)	The maximum amount of memory (in megabytes) that Intelligence Server can use to generate a report or document in HTML. If this limit is reached, the HTML document is not generated and the user sees an error message.	*Limiting the memory consumption for file generation*

Governing Rules: Default: Memory Settings category in Intelligence Server configuration

Governor	Description	See page
Enable Web request job throttling	A check box that enables the following governors: • **Maximum Intelligence Server use of total memory** • **Minimum machine free physical memory**	*Governing memory for requests from MicroStrategy Web products*
Maximum Intelligence Server use of total memory (%)	The maximum amount of total system memory (RAM + Page File) that can be used by the Intelligence Server process (`MSTRSVR.exe`) compared to the total amount of memory on the machine. If the limit is met, all requests from MicroStrategy Web products of any nature (log in, report execution, search, folder browsing) are denied until the conditions are resolved.	*Maximum Intelligence Server use of total memory sets the maximum amount of total system memory (RAM plus Page File) that can be used by the Intelligence Server process compared to the total amount of memory on the machine.*
Minimum machine free physical memory (%)	The minimum amount of physical memory (RAM) that needs to be available, as a percentage of the total amount of physical memory on the machine. If the limit is met, all requests from MicroStrategy Web products (e.g., log in, report execution, search, folder browsing) are denied until the conditions are resolved.	*Minimum machine free physical memory sets the minimum amount of RAM that must remain available for Web requests. This value is a percentage of the total amount of physical memory on the machine, not including the Page File memory.*

 © 2016, MicroStrategy Inc.

Governor	Description	See page
Enable single memory allocation governing	A check box that enables the **Maximum single allocation size** governor.	*Using the Memory Contract Manager*
Maximum single allocation size (MBytes)	Prevents Intelligence Server from granting a request that would exceed this limit.	*Using the Memory Contract Manager*
Enable memory contract management	A check box that enables the following governors: • **Minimum reserved memory (MB or %)** • **Maximum use of virtual address space (%)** • **Memory request idle time**	*Using the Memory Contract Manager*
Minimum reserved memory (MBytes or %)	The amount of system memory, in either MB or a percent, that must be reserved for processes external to Intelligence Server.	*Using the Memory Contract Manager*
Maximum use of virtual address space (%)	The maximum percent of the process' virtual address space that Intelligence Server can use before entering memory request idle mode. This setting is used in 32-bit operating systems and is no longer applicable. In 64-bit operating systems, to control the amount of memory available for Intelligence Server, use the Minimum reserved memory governor.	*Using the Memory Contract Manager*
Memory request idle time (sec)	The amount of time Intelligence Server denies requests that may result in memory depletion. If Intelligence Server does not return to acceptable memory conditions before the idle time is reached, Intelligence Server shuts down and restarts.	*Using the Memory Contract Manager*
Temporary Storage Setting: Maximum RAM for Working Set cache (MB)	The maximum amount of memory that can be used for report instances referenced by messages in the Working Set.	*Working set (MicroStrategy Web only)*

Governing Rules: Default: Temporary Storage Settings category in Intelligence Server configuration

Governor	Description	See page
Working Set file directory	The location where the user's active working sets are written to disk if they have been forced out of the pool of memory allocated for the Maximum RAM for working set cache. The default is .\TmpPool	*Governing saved user session information (MicroStrategy Web only)*

Governor	Description	See page
Session Recovery and Deferred Inbox storage directory	Specifies the where the session information is written to disk. The default is .\TmpPool	*Governing saved user session information (MicroStrategy Web only)*
Enable Web User Session Recovery on Logout	If selected, allows Web users to recover their sessions.	*Governing saved user session information (MicroStrategy Web only)*
Session Recovery backup expiration (hrs)	How many hours a session backup can remain on disk before it is considered expired. After it is expired, the user cannot recover the session.	*Governing saved user session information (MicroStrategy Web only)*

Governing Rules: Default: Import Data category in Intelligence Server configuration

Governor	Description	See page
Number of connections by priority	The number of connection threads to create for Import Data jobs, depending on whether the priority of the job is high, medium or low. You must determine the number of threads that quickly serves users without overloading the system.	*Managing database connection threads*

Governing Rules: Default: Catalog cache category in Intelligence Server configuration

Governor	Description	See page
Enable catalog cache	A check box that enables the **Maximum use of memory (MB)** governor.	
Maximum use of memory (MB)	Limits the maximum amount of memory, in megabytes, used by the catalog cache. The default value is 25 MB	

History Settings: General category in Intelligence Server configuration

Governor	Description	See page
Maximum number of messages per user	The maximum number of History messages that can exist in a user's History List at any time. When the limit is reached, the oldest message is removed.	*Saving report results: History List*
Message lifetime (days)	The length of time before a History List message expires and is automatically deleted. A value of -1 indicates that messages do not expire.	*Saving report results: History List*
Repository type	Select **File Based** for History List messages to be stored on disk in a file system, or **Database Based** for History List messages to be stored in a database (recommended).	*Saving report results: History List*

 © 2016, MicroStrategy Inc.

Project Configuration Editor

These governors can be set per project. To access them, right-click the project, select **Project Configuration**, then select the category as noted below.

Project definition: Advanced category

Governor	Description	See page
Maximum number of elements to display	The maximum number of attribute elements that can be being retrieved from the data warehouse at one time.	*Limiting the number of elements displayed and cached at a time*

Governing Rules: Default: Result sets category in Project Configuration

Governor	Description	See page
Intelligence Server Elapsed Time - Interactive reports (sec)	The amount of time that an ad-hoc report request can take before it is canceled. This includes time spent resolving prompts, waiting for autoprompts, waiting in the job queue, executing SQL, analytical calculation, and preparing report results.	*Limiting the maximum report execution time*
Intelligence Server Elapsed Time - Scheduled reports (sec)	The amount of time that a scheduled report request can take before it is canceled. This includes time spent resolving prompts, waiting for autoprompts, waiting in the job queue, executing SQL, analytical calculation, and preparing report results.	*Limiting the maximum report execution time*
Final Result Rows - Intelligent Cubes	The maximum number of rows that can be returned to Intelligence Server for an Intelligent Cube request. This setting is applied by the Query Engine when retrieving the results from the database. This is the default for all reports in a project and can be overridden for individual reports by using the VLDB settings.	*Limiting the number of report result rows*
Final Result Rows - Data marts	The maximum number of rows that can be returned to Intelligence Server for a data mart report request. This setting is applied by the Query Engine when retrieving the results from the database. This is the default for all reports in a project and can be overridden for individual reports by using the VLDB settings.	*Limiting the number of report result rows*
Final Result Rows - All other reports	The maximum number of rows that can be returned to Intelligence Server for a standard report request. This setting is applied by the Query Engine when retrieving the results from the database. This is the default for all reports in a project and can be overridden for individual reports by using the VLDB settings.	*Limiting the number of report result rows*
All intermediate result rows	The maximum number of rows that can be in an intermediate result set used for analytical processing in Intelligence Server. This is the default for all reports in a project and can be overridden by using the VLDB settings for individual reports.	*Limiting the number of intermediate rows*
All element browsing result rows	The maximum number of rows that can be retrieved from the data warehouse for an element request.	*Limiting the number of element rows*
Memory consumption during SQL generation (MB)	The maximum amount of memory (in megabytes) that Intelligence Server can use for SQL generation. The default is -1, which indicates no limit.	*Limiting a report's SQL per pass*

© 2016, MicroStrategy Inc.

Governor	Description	See page
Memory consumption during data fetching (MB)	The maximum amount of memory (in megabytes) that Intelligence Server can use for importing data. The default is 2048 MB (2 GB).	*Limiting the memory used during data fetching*

Governing Rules: Default: Jobs category in Project Configuration

Governor	Description	See page
Jobs per user account	The maximum number of concurrent jobs per user account and project.	*Limiting the number of jobs per user session and per user account*
Jobs per user session	The maximum number of concurrent jobs a user can have during a session.	*Limiting the number of jobs per user session and per user account*
Executing jobs per user	The maximum number of concurrent jobs a single user account can have executing in the project. If this condition is met, additional jobs are placed in the queue until executing jobs finish.	*Limiting the number of executing jobs per user and project*
Jobs per project - interactive	The maximum number of concurrent ad-hoc jobs that the project can process at a time.	*Limiting the number of jobs per project*
Jobs per project - scheduled	The maximum number of concurrent scheduled jobs that the project can process at a time.	*Limiting the number of jobs per project*

Governing Rules: Default: User sessions category in Project Configuration

Governor	Description	See page
User sessions per project	The maximum number of user sessions that are allowed in the project. When the limit is reached, users other than the Administrator cannot log in.	*Limiting user sessions per project*
Concurrent interactive project sessions per user	The maximum number of concurrent sessions per user.	*Limiting user sessions per project*

Governing Rules: Default: Subscriptions category in Project Configuration

Governor	Description	See page
Maximum History List subscriptions per user	The maximum number of reports or documents to which a user can be subscribed for delivery to the History List.	*Managing subscriptions*

 © 2016, MicroStrategy Inc.

Governor	Description	See page
Maximum Cache Update subscriptions per user	The maximum number of reports or documents to which a user can be subscribed for updating caches.	*Managing subscriptions*
Maximum email subscriptions per user	The maximum number of reports or documents to which a user can be subscribed for delivery to an email address (Distribution Services only).	*Managing subscriptions*
Maximum file subscriptions per user	The maximum number of reports or documents to which a user can be subscribed for delivery to a file location (Distribution Services only).	*Managing subscriptions*
Maximum print subscriptions per user	The maximum number of reports or documents to which a user can be subscribed for delivery to a printer (Distribution Services only).	*Managing subscriptions*
Maximum Mobile subscriptions per user	The maximum number of reports or documents to which a user can be subscribed for delivery to a Mobile device (MicroStrategy Mobile only).	*Managing subscriptions*

Governing Rules: Default: Import Data category in Project Configuration

Governor	Description	See page
Maximum file size (MB)	The maximum size for a file to be imported for use as a data source. Files larger that this value cannot be opened during data import.	*Defining limits for Intelligent Cubes created using the Import Data feature*
Maximum quota per user (MB)	The maximum size of all data import cubes for each individual user.	*Defining limits for Intelligent Cubes created using the Import Data feature*

Caching: Result Caches: Storage category in Project Configuration

Governor	Description	See page
Datasets - Maximum RAM usage (MBytes)	The maximum amount of memory reserved for the creation and storage of report and dataset caches. This setting should be configured to at least the size of the largest cache file, or that report will not be cached.	*Configuring result cache settings*
Datasets - Maximum number of caches	The maximum number of report and dataset caches that the project can have at a time.	*Managing result caches*
Formatted Documents - Maximum RAM usage (MBytes)	The maximum amount of memory reserved for the creation and storage of document caches. This setting should be configured to be at least the size of the largest cache file, or that report will not be cached.	*Configuring result cache settings*

© 2016, MicroStrategy Inc.

Governor	Description	See page
Formatted Documents - Maximum number of caches	The maximum number of document caches that the project can have at a time.	*Managing result caches*
RAM swap multiplier	The amount of memory that is swapped to disk, relative to the size of the cache being swapped into memory. For example, if the **RAM swap multiplier** setting is 2 and the requested cache is 80 Kbytes, 160 Kbytes are swapped from memory to disk.	*RAM swap multiplier*

Caching: Result caches: Maintenance category in Project Configuration

Governor	Description	See page
Never expire caches	Determines whether or not caches automatically expire.	*Never expire caches*
Cache duration (Hours)	The amount of time that a result cache remains valid.	*Cache duration (Hours)*
Do not Apply Automatic Expiration Logic for reports containing dynamic dates	Select this check box for report caches with dynamic dates to expire in the same way as other report caches.	*Cache expiration and dynamic dates*

Caching: Auxiliary Caches: Objects category in Project Configuration

Governor	Description	See page
Server - Maximum RAM usage (MBytes)	The amount of memory that Intelligence Server allocates for object caching.	*Summary table of object caching settings*
Client - Maximum RAM usage (MBytes)	The amount of memory that Developer allocates for object caching.	*Summary table of object caching settings*

Caching: Auxiliary Caches: Elements category in Project Configuration

Governor	Description	See page
Server - Maximum RAM usage (MBytes)	The amount of memory that Intelligence Server allocates for element caching.	*Summary table of element cache settings*
Client - Maximum RAM usage (MBytes)	The amount of memory that Developer allocates for object caching.	*Summary table of element cache settings*

Caching: Subscription Execution category in Project Configuration

 © 2016, MicroStrategy Inc.

Governor	Description	See page
Re-run history list and mobile subscriptions against the warehouse	Causes new subscriptions to create caches or update existing caches by default when a report or document is executed and that report/document is subscribed to the History List or a Mobile device.	*Managing scheduled administration tasks*
Re-run file, email, print, or ftp subscriptions against the warehouse	Causes new subscriptions to create caches or update existing caches by default when a report or document is executed and that report/document is subscribed to a file, email, or print device.	*Managing scheduled administration tasks*
Do not create or update matching caches	Prevents subscriptions from creating or updating caches by default.	*Managing scheduled administration tasks*
Keep document available for manipulation for History List subscriptions only	Retains a document or report for later manipulation that was delivered to the History List.	*Managing scheduled administration tasks*

Intelligent Cubes: General category in Project Configuration

Governor	Description	See page
Maximum RAM Usage (MBytes)	The maximum amount of memory used on Intelligence Server by Intelligent Cubes for this project.	*Defining memory limits for Intelligent Cubes*
Maximum number of cubes	The maximum number of Intelligent Cubes that can be loaded onto Intelligence Server for this project.	*Defining memory limits for Intelligent Cubes*

Database connection

This set of governors can be set by modifying a project source's database instance and then modifying either the number of Job Prioritization connections or the Database connection. For more details on each governor, see the page references in the table below.

ODBC Settings

Governor	Description	See page
Number of database connection threads	The total number of High, Medium, and Low database connections that are allowed at a time between Intelligence Server and the data warehouse (set on the database instance's **Job Prioritization** tab).	*Managing database connection threads*
Maximum cancel attempt time (sec)	The maximum amount of time that the Query Engine waits for a successful attempt to cancel a query.	*Maximum cancel attempt time*

© 2016, MicroStrategy Inc.

Governor	Description	See page
Maximum query execution time (sec)	The maximum amount of time that a single pass of SQL may execute on the data warehouse.	*Maximum query execution time*
Maximum connection attempt time (sec)	The maximum amount of time that Intelligence Server waits to connect to the data warehouse.	*Maximum connection attempt time*

Database Connection Caching

Governor	Description	See page
Connection lifetime (sec)	The amount of time that an active database connection thread remains open and cached on Intelligence Server.	*Connection lifetime*
Connection idle timeout (sec)	The amount of time that an inactive database connection thread remains cached until it is terminated.	*Connection idle timeout*

VLDB settings

These settings can be changed in the VLDB Properties dialog box for either reports or the database instance. For information about accessing these properties, see the page reference for each property in the table below. For complete details about all VLDB properties, see the *VLDB Properties* chapter in the Supplemental Reference for System Administration.

Governor	Description	See page
Intermediate row limit	The maximum number of rows that can be in an intermediate table used by Intelligence Server. This setting overrides the project's default **Number of intermediate result rows** setting	*Limiting the number of intermediate rows*
Results Set Row Limit	The maximum number of rows that can be in a report result set. This setting overrides the project's default **Number of report result rows** set.	*Limiting the number of report result rows*
SQL time out (per pass)	The amount of time, in seconds, that any SQL pass can execute on the data warehouse. This can be set at the database instance and report levels.	*SQL Time Out (Per Pass) (database instance and report)*
Maximum SQL size	The maximum size (in bytes) that the SQL statement can be. This can be set at the database instance level.	*Maximum SQL Size (database instance)*

© 2016, MicroStrategy Inc.

Tuning Narrowcast Server and Intelligence Server

If you are using Narrowcast Server as part of your system to deliver reports to users, you should be aware of its impact on Intelligence Server system resources. This section includes relevant discussions about:

- How you design Narrowcast Server applications (*Application design considerations, page 367*)
- How Narrowcast Server connects to Intelligence Server (*How Narrowcast Server connects to Intelligence Server, page 368*)

For more information, refer to the *Narrowcast Server System Administrator Guide*.

Application design considerations

Depending on how you design applications in Narrowcast Server you can place more or less load on Intelligence Server. Two main options to consider are personal report execution and personal page execution.

Personal report execution (PRE) executes a separate report for each set of users with unique personalization. Users can have reports executed under the context of the corresponding Intelligence Server user if desired. Using this option, security profiles defined in Developer are maintained. However if the system contains many users who all have unique personalization, this option can place a large load on Intelligence Server.

Personal page execution (PPE) executes one multi-page report for all users in a segment and then uses this single report to provide personalized content (pages) for different users. All users have their reports executed under the context of the same Intelligence Server user, so individual security profiles are not maintained. However, the load on Intelligence Server may be significantly lower than for PRE in some cases.

For more detailed information about these options, refer to the *Narrowcast Server Application Designer Guide*, specifically the chapter on Page Personalization and Dynamic Subscriptions.

Two additional points to consider in designing your Narrowcast Server applications are:

- Timing of Narrowcast Server jobs: You can schedule reports to run at off-peak hours when Intelligence Server's load from MicroStrategy Web products and Developer users is lowest.
- Intelligence Server selection: You can send Narrowcast Server jobs to a specific Intelligence Server to ensure that some Intelligence Servers are used solely for MicroStrategy Web products or Developer.

© 2016, MicroStrategy Inc.

How Narrowcast Server connects to Intelligence Server

Narrowcast Server can connect to a specific Intelligence Server. Narrowcast Server does this by using one or more information sources to point to and connect to the desired Intelligence Servers. Note the following points:

- Intelligence Server provides automatic load balancing for Narrowcast Server requests. Once an information source is configured, jobs using that information source go to the appropriate Intelligence Server for the most efficient response.
- Narrowcast Server can connect to any Intelligence Server in a cluster—this does not need to be the primary node.
- You can balance the load manually by creating multiple information sources or by using a single information source pointing to one Intelligence Server, thereby designating it to handle all Narrowcast Server requests.

 © 2016, MicroStrategy Inc.

9

CLUSTERING MULTIPLE MICROSTRATEGY SERVERS

Introduction

A clustered set of machines provides a related set of functionality or services to a common set of users. MicroStrategy recommends clustering Intelligence Servers in environments where access to the data warehouse is mission-critical and system performance is of utmost importance. Intelligence Server provides you the functionality to cluster a group of Intelligence Server machines to take advantage of the many benefits available in a clustered environment.

This section provides the following information:

- *Overview of clustering, page 370*
- *The clustered architecture, page 372*
- *Prerequisites for clustering Intelligence Servers, page 377*
- *Clustering Intelligence Servers, page 380*
- *Managing your clustered system, page 390*
- *Connecting MicroStrategy Web to a cluster, page 397*

© 2016, MicroStrategy Inc.

Overview of clustering

A cluster is a group of two or more servers connected to each other in such a way that they behave like a single server. Each machine in the cluster is called a node. Because each machine in the cluster runs the same services as other machines in the cluster, any machine can stand in for any other machine in the cluster. This becomes important when one machine goes down or must be taken out of service for a time. The remaining machines in the cluster can seamlessly take over the work of the downed machine, providing users with uninterrupted access to services and data.

You can cluster MicroStrategy components at two levels:

- You can cluster Intelligence Servers using the built-in Clustering feature. A Clustering license allows you to cluster up to four Intelligence Server machines. For instructions on how to cluster Intelligence Servers, see *Clustering Intelligence Servers, page 380*.
- You can cluster MicroStrategy Web servers using third-party clustering software, such as Cisco Local Router, Microsoft Windows Load Balancing Service, or Microsoft Network Load Balancing. Most clustering tools work by using IP distribution based on the incoming IP addresses. For details on implementing this clustering method, see the documentation for your third-party clustering software.

The built-in clustering feature allows you to connect MicroStrategy Web to a cluster of Intelligence Servers. For instructions, see *Connecting MicroStrategy Web to a cluster, page 397*.

Benefits of clustering

Clustering Intelligence Servers provides the following benefits:

- Increased resource availability: If one Intelligence Server in a cluster fails, the other Intelligence Servers in the cluster can pick up the workload. This prevents the loss of valuable time and information if a server fails.
- Strategic resource usage: You can distribute projects across nodes in whatever configuration you prefer. This reduces overhead because not all machines need to be running all projects, and allows you to use your resources flexibly.
- Increased performance: Multiple machines provide greater processing power.
- Greater scalability: As your user base grows and report complexity increases, your resources can grow.
- Simplified management: Clustering simplifies the management of large or rapidly growing systems.

Clustering enables you to implement the following strategies in your business intelligence environment, all of which are discussed in this chapter:

- Failover support (see *Failover support, page 371*)

 © 2016, MicroStrategy Inc.

- Load balancing (see *Load balancing, page 371*)
- Project distribution and project failover (see *Project distribution and project failover, page 371*)

Failover support

Failover support ensures that a business intelligence system remains available for use if an application or hardware failure occurs. Clustering provides failover support in two ways:

- Load redistribution: When a node fails, the work for which it is responsible is directed to another node or set of nodes.
- Request recovery: When a node fails, the system attempts to reconnect MicroStrategy Web or Web Universal users with queued or processing requests to another node. Users must log in again to be authenticated on the new node. The user is prompted to resubmit job requests.

Load balancing

Load balancing is a strategy aimed at achieving even distribution of user sessions across Intelligence Servers, so that no single machine is overwhelmed. This strategy is especially valuable when it is difficult to predict the number of requests a server will receive. MicroStrategy achieves four-tier load balancing by incorporating load balancers into the MicroStrategy Web and Web Universal products.

Load is calculated as the number of user sessions connected to a node. The load balancers collect information on the number of user sessions each node is carrying. Using this information at the time a user logs in to a project, MicroStrategy Web or Web Universal connects them to the Intelligence Server node that is carrying the lightest session load. All requests by that user are routed to the node to which they are connected until the user disconnects from the MicroStrategy Web product.

Project distribution and project failover

When you set up several server machines in a cluster, you can distribute projects across those clustered machines or nodes in any configuration, in both Windows and UNIX/Linux environments. All servers in a cluster do not need to be running all projects. Each node in the cluster can host a different set of projects, which means only a subset of projects need to be loaded on a specific Intelligence Server machine. This feature provides you with flexibility in using your resources, and it provides better scalability and performance because of less overhead on each Intelligence Server machine.

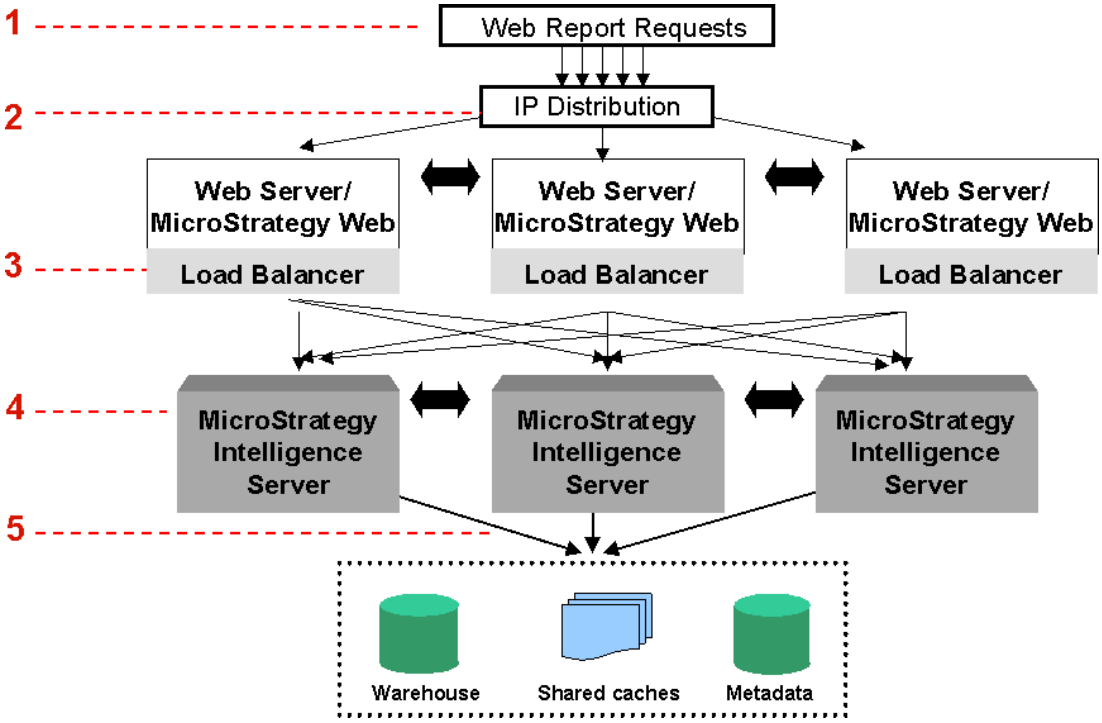
Distributing projects across nodes also provides project failover support. For example, one server is hosting project A and another server is hosting projects B and C. If the first server fails, the other server can host all three projects to ensure project availability.

Project creation, duplication, and deletion in a three-tier, or server, connection are automatically broadcast to all nodes during runtime to ensure synchronization across the cluster.

© 2016, MicroStrategy Inc.

The clustered architecture

The diagram below shows report distribution in a four-tier clustered environment. The clustered Intelligence Servers are shown in gray.



The node of the cluster that performs all job executions is the node that the client application, such as Developer, connects to. This is also the node that can be monitored by an administrator using the monitoring tools.

The following steps describe a typical job process in a clustered, four-tier environment. They correspond to the numbers in the report distribution flow diagram above.

1. MicroStrategy Web or Web Universal users log in to a project and request reports from their Web browsers.
2. A third-party IP distribution tool such as Cisco Local Router, Microsoft Network Load Balancing, or Microsoft Windows Load Balancing Service distributes the user connections from the MicroStrategy Web clients among web servers.
3. The MicroStrategy Web product load balancers on each server collect load information from each cluster node and then connect the users to the nodes that carry the lightest loads and that run the project the user requested. All report requests are then processed by the nodes to which the users are connected.
4. The Intelligence Server nodes receive the requests and process them. In addition, the nodes communicate with each other to maintain metadata synchronization and cache accessibility across nodes.
5. The nodes send the requests to the warehouse as queries.

© 2016, MicroStrategy Inc.

Query flow in a clustered environment is identical to a standard query flow in an unclustered environment (see *Processing jobs, page 32*), with two exceptions:

- Result (report and document) caches and Intelligent Cubes: When a query is submitted by a user, if an Intelligent Cube or a cached report or document is not available locally, the server will retrieve the cache (if it exists) from another node in the cluster. For an introduction to report and document caching, see *Result caches, page 400*. For an introduction to Intelligent Cubes, see *Chapter 11, Managing Intelligent Cubes*.
- History Lists: Each user's History List, which is held in memory by each node in the cluster, contains direct references to the relevant cache files. Accessing a report through the History List bypasses many of the report execution steps, for greater efficiency. For an introduction to History Lists, see *Saving report results: History List, page 425*.

Synchronizing cached information across nodes in a cluster

In a clustered environment, each node shares cached information with the other nodes so that the information users see is consistent regardless of the node to which they are connected when running reports. All nodes in the cluster synchronize the following cached information:

- Metadata information and object caches (for details, see *Synchronizing metadata, page 373*)
- Result caches and Intelligent Cubes (for details, see *Sharing result caches and Intelligent Cubes in a cluster, page 374*)
- History Lists (for details, see *Synchronizing History Lists, page 377*)

To view clustered cache information, such as cache hit counts, use the Cache Monitor.

Result cache settings are configured per project, and different projects may use different methods of result cache storage. Different projects may also use different locations for their cache repositories. However, History List settings are configured per project source. Therefore, different projects cannot use different locations for their History List backups.

For result caches and History Lists, you must configure either multiple local caches or a centralized cache for your cluster. The following sections describe the caches that are affected by clustering, and it presents the procedures to configure caches across cluster nodes.

Synchronizing metadata

Metadata synchronization refers to the process of synchronizing object caches across all nodes in the cluster.

For example, when a user connected to a node in a cluster modifies a metadata object, the cache for that object on other nodes is no longer valid. The node that processed the

change automatically notifies all other nodes in the cluster that the object has changed. The other nodes then delete the old object cache from memory. The next request for that object that is processed by another node in the cluster is executed against the metadata, creating a new object cache on that node.

In addition to server object caches, client object caches are also invalidated when a change occurs. When a user requests a changed object, the invalid client cache is not used and the request is processed against the server object cache. If the server object cache has not been refreshed with the changed object, the request is executed against the metadata.

Sharing result caches and Intelligent Cubes in a cluster

In a non-clustered environment, Intelligent Cubes and report and document caches (result caches) are typically stored on the Intelligence Server machine. For an overview of Intelligent Cubes, see *Chapter 11, Managing Intelligent Cubes*, or see the *MicroStrategy In-memory Analytics Guide*. For an overview of result caches, see *Result caches, page 400*.

In a clustered environment, each node in a cluster must share its result caches and Intelligent Cubes with the other nodes, so all clustered machines have the latest cache information. For example, for a project, result caches on each node that has loaded the project are shared among other nodes in the cluster that have also loaded the project. Configuring caches to be shared among appropriate nodes eliminates the overhead associated with executing the same report or document on multiple nodes.

Note the following:

- Both memory and disk caches are shared among nodes.
- When an Intelligent Cube is updated, either through Incremental Refresh or by republishing the Intelligent Cube, the updated Intelligent Cube is available on all nodes of the cluster as soon as it is loaded into memory.

Intelligent Cube and result cache sharing among nodes can be configured in one of the following ways:

- **Local caching**: Each node hosts its own cache file directory and Intelligent Cube directory. These directories need to be shared so that other nodes can access them. For more information, see *Local caching, page 375*.

  If you are using local caching, the cache directory must be shared as "ClusterCaches" and the Intelligent Cube directory must be shared as "ClusterCube". These are the share names Intelligence Server looks for on other nodes to retrieve caches and Intelligent Cubes.

- **Centralized caching**: All nodes have the cache file directory and Intelligent Cube directory set to the same network locations, `\\<machine name>\<shared cache folder name>` and `\\<machine name>\<shared Intelligent Cube folder name>`. For more information, see *Centralized caching, page 376*.

 © 2016, MicroStrategy Inc.

For caches on Windows machines, and on UNIX/Linux machines using Samba, set the path to `\\<machine name>\<shared cache folder name>`. For caches on UNIX/Linux machines, set the path to `//<SharedLocation>/<CacheFolder>`.

The following table summarizes the pros and cons of the result cache configurations:

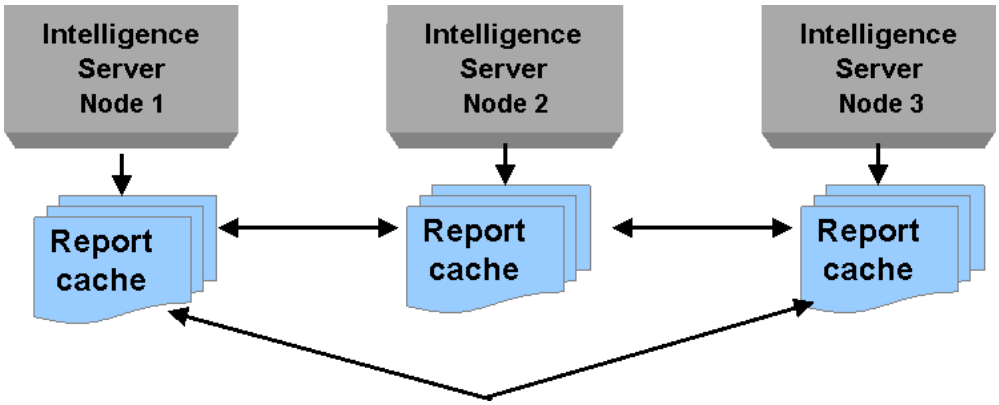
	Pros	Cons
Local caching	• Allows faster read and write operations for cache files created by the local server. • Faster backup of cache lookup table. • Allows most caches to remain accessible even if one node in a cluster goes offline.	• The local cache files may be temporarily unavailable if an Intelligence Server is taken off the network or powered down. • A document cache on one node may depend on a dataset that is cached on another node, creating a multi-node cluster dependency.
Centralized caching	• Allows for easier backup process. • Allows all cache files to be accessible even if one node in a cluster goes offline. • May better suit some security plans because nodes using a network account are accessing only one machine for files.	• All cache operations are required to go over the network if shared location is not on one of the Intelligence Server machines. • Requires additional hardware if shared location is not on an Intelligence Server. • All caches become inaccessible if the machine hosting the centralized caches goes offline.

MicroStrategy recommends storing the result caches locally if your users mostly do ad hoc reporting. In ad hoc reporting the caches are not used very much, and the overhead incurred by creating the caches on a remote file server outweighs the low probability that a cache may be used. On the other hand, if the caches are to be heavily used, centralized caching may suit your system better.

For steps to configure cache files with either method, see *Configuring caches in a cluster, page 380*.

Local caching

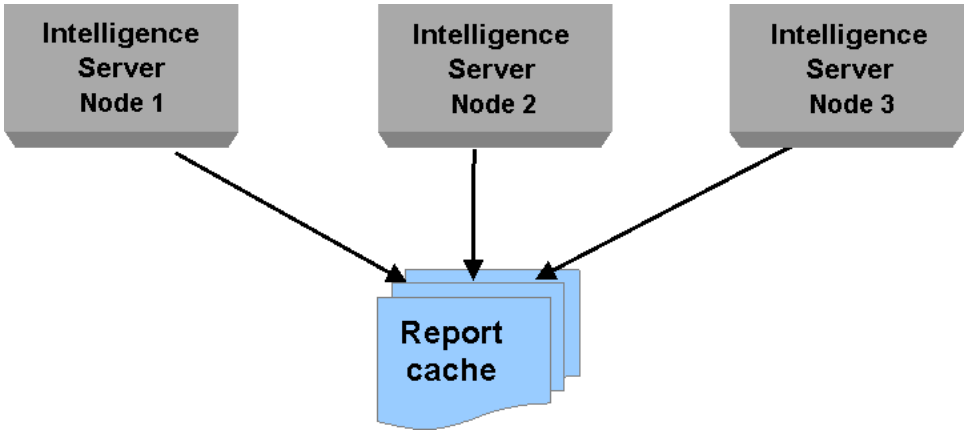
In this cache configuration, each node maintains its own local Intelligent Cubes and local cache file and, thus, maintains its own cache index file. Each node's caches are accessible by other nodes in the cluster through the cache index file. This is illustrated in the diagram below.



For example, User A, who is connected to node 1, executes a report and thus creates report cache A on node 1. User B, who is connected to node 2, executes the report. Node 2 checks its own cache index file first. When it does not locate report cache A in its own cache index file, it checks the index file of other nodes in the cluster. Locating report cache A on node 1, it uses that cache to service the request, rather than executing the report against the warehouse.

Centralized caching

In this cache configuration, all nodes in the cluster use one shared, centralized location for Intelligent Cubes and one shared, centralized cache file location. These can be stored on one of the Intelligence Server machines or on a separate machine dedicated to serving the caches. The Intelligent Cubes, History List messages, and result caches for all the Intelligence Server machines in the cluster are written to the same location. In this option, only one cache index file is maintained. This is illustrated in the diagram below.



For example, User A, who is connected to node 1, executes report A and thus creates report cache A, which is stored in a centralized file folder. User B, who is connected to node 2, executes report A. Node 2 checks the centralized cache index file for report cache A. Locating report cache A in the centralized file folder, it uses that cache to service the request, regardless of the fact that node 1 originally created the cache.

 © 2016, MicroStrategy Inc.

Synchronizing History Lists

A History List is a set of pointers to cache files. Each user has his or her own History List, and each node in a cluster stores the pointers created for each user who is connected to that node. Each node's History List is synchronized with the rest of the cluster. Even if report caching is disabled, History List functionality is not affected.

If you are using a database-based History List, History List messages and their associated caches are stored in the database and automatically synchronized across all nodes in the cluster.

If you are using a file-based History List, the Intelligence Server Inbox folder contains the collection of History List messages for all users, which appear in the History folder in Developer. Inbox synchronization refers to the process of synchronizing History Lists across all nodes in the cluster, so that all nodes contain the same History List messages. Inbox synchronization enables users to view the same set of personal History List messages, regardless of the cluster node to which they are connected.

For more background information on History Lists, see *Saving report results: History List, page 425*. For steps to set up History List sharing in a file-based system, see *Configuring History List sharing using multiple local cache files, page 383*.

MicroStrategy recommends that you enable user affinity clustering to minimize History List resource usage. User affinity clustering causes Intelligence Server to connect all sessions for a user to the same node of the cluster. This enables Intelligence Server to keep the user's History List on one node of the cluster. Resource use is minimized because the pointers to the History List are not stored on multiple machines. In addition, if you are using a file-based History List, the History List is never out of sync across multiple nodes of the cluster. For instructions on how to enable user affinity clustering, see *Configuring History Lists in a clustered environment, page 386*.

Prerequisites for clustering Intelligence Servers

Before you can cluster Intelligence Servers in your system, you must fulfil these prerequisites.

MicroStrategy prerequisites

- You must have purchased an Intelligence Server license that allows clustering. To determine the license information, use the License Manager tool and verify that the Clustering feature is available for Intelligence Server. For more information on using License Manager, see *Chapter 5, Managing Your Licenses*.
- The computers to be clustered must all have the same version of Intelligence Server installed.
- All MicroStrategy projects on the clustered machines must be based on the same metadata.

© 2016, MicroStrategy Inc.

- At least one project must be defined in the metadata.
- No more than one Intelligence Server can be configured for a single machine. Multiple instances of Intelligence Server should not run on the same machine for clustering purposes.
- The user account under which the Intelligence Server service is running must have full control of cache and History List folders on all nodes. Otherwise, Intelligence Server will not be able to create and access cache and History List files.
- Server definitions store Intelligence Server configuration information. MicroStrategy strongly recommends that all servers in the cluster use the same server definition. This ensures that all nodes have the same governing settings.

  Server definitions can be modified from Developer through the Intelligence Server Configuration Editor and the Project Configuration Editor. For instructions, see the *MicroStrategy Developer Help*.
- Developer must be installed on a Windows machine to administer the cluster. This version of Developer must be the same as the version of Intelligence Servers. For example, if the Intelligence Servers are running MicroStrategy Analytics Enterprise, Developer must also be Analytics Enterprise.
- You must have access to the Cluster view of the System Administration monitor in Developer. Therefore, you must have the Administration privilege to create a cluster. For details about the Cluster view of the System Administration monitor, see *Managing your clustered system, page 390*.
- The computers that will be clustered must have the same intra-cluster communication settings. To configure these settings, on each Intelligence Server machine, in Developer, right-click the project source and select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens. Under the **Server definition** category, select **General**. For instructions, see the *MicroStrategy Developer Help*.
- The same caching method (localized or centralized caching) should be used for both result caches and file-based History Lists. For information about localized and centralized caching, see *Sharing result caches and Intelligent Cubes in a cluster, page 374*.

Server prerequisites

- The machines to be clustered must be running the same version of the same operating system. For example, you cannot cluster two machines when one is running on Windows 2008 and one is running on Windows 2003.
- Load balancing and system configuration are simpler if identical hardware is used for each of the clustered nodes.
- If you are using time-based schedules in a clustered environment, all the nodes in the cluster must have their clocks synchronized.
- The RDBMS containing the metadata and warehouse instances must already be set up on machines separate from the Intelligence Server nodes.

 © 2016, MicroStrategy Inc.

- Information on the clustered configuration is stored in the metadata, so the machines to be clustered must use the same metadata repository. The metadata may be created from any of the nodes, and it needs to be set up only once. When you create or modify the server definition in the MicroStrategy Configuration Wizard, you can specify either a new or an existing metadata repository for Intelligence Server to use.
- The required data source names (DSNs) must be created and configured for Intelligence Server on each machine. MicroStrategy strongly recommends that you configure both servers to use the same metadata database, warehouse, port number, and server definition.
- All nodes must join the cluster before you make any changes to any governing settings, such as in the Intelligence Server Configuration Editor.

Prerequisites for Windows clustering

- On all machines to be clustered, each network card must be configured to enable Netbios over TCP/IP. Otherwise, cache sharing is not possible using Netbios names (ClusterCaches, ClusterCube, and ClusterInbox).
- When Intelligence Server is installed, the last step is to choose a user identity under which the service will run. To run a clustered configuration, the user must be a domain account that has a trust relationship with each of the computers in the cluster. This allows resources to be shared across the network.
- The service user's Regional Options settings must be the same as the clustered system's Regional Options settings.

Prerequisites for UNIX/Linux clustering

- MicroStrategy strongly recommends that all servers in a cluster use the same server definition. Therefore, in some cases you cannot specify the cache location with an absolute path such as `/<machine_name>`. This occurs because the location would have to be different for each server machine. To solve this problem, use relative paths and soft links. A soft link is a special type of UNIX file that refers to another file by its path name. A soft link is created with the `ln (link)` command:

  `ln -s` *`OLDNAME NEWNAME`*

  where

  `OLDNAME` is the target of the link, usually a path name.

  `NEWNAME` is the path name of the link itself.

  Most operations (open, read, write) on the soft link automatically de-reference it and operate on its target (OLDNAME). Some operations (for example, removing) work on the link itself (NEWNAME).
- Confirm that each server machine works properly, and then shut each down.

Clustering Intelligence Servers

Below is a high-level overview of the steps to cluster Intelligence Servers:

1. Confirm that you have fulfilled the prerequisites for clustering Intelligence Servers.

   For a list of these prerequisites, see *Prerequisites for clustering Intelligence Servers, page 377*.

2. Configure the caches to synchronize information across nodes.

   Before Intelligence Servers can be clustered, the information such as report caches and History Lists must be synchronized among them. For steps, see *Configuring caches in a cluster, page 380*.

3. Join nodes.

   You create a cluster by joining Intelligence Servers that have been synchronized. For steps, see *Joining the nodes in a cluster, page 386*.

4. Test the clustered system.

   Once the cluster has been created, you should test it to make sure that the caches and metadata are being shared properly among nodes. For steps on performing various tests, see *Verifying the clustered system is working, page 387*.

5. (Optional) Distribute projects across nodes.

   Once the cluster has been created, you may wish to distribute your projects across the cluster to improve performance. For more information and steps, see *Distributing projects across nodes in a cluster, page 388*.

Configuring caches in a cluster

You can configure caches in one of two ways:

- **Local caching**: Each node hosts its own cache file directory and Intelligent Cube directory. These directories need to be shared so that other nodes can access them. For more information, see *Local caching, page 375*.
- **Centralized caching**: All nodes have the cache file directory and Intelligent Cube directory set to the same network locations. For more information, see *Centralized caching, page 376*.

For steps to configure caches in either way, follow the instructions below depending on your operating system:

- *Configuring caches in a cluster on Windows, page 381*
- *Configuring caches in a cluster on UNIX/Linux, page 383*

 © 2016, MicroStrategy Inc.

Configuring caches in a cluster on Windows

Use one of the procedures below to share cache files among the nodes in your cluster. MicroStrategy strongly recommends that each node in your cluster use the same server definition. In this case, you need to configure the cache location in Intelligence Server only one time. However, you must create the shared folders on each node separately. For a detailed explanation of the two methods of cache sharing, see *Sharing result caches and Intelligent Cubes in a cluster, page 374*.

To configure cache sharing using multiple local cache files

1. Open the Project Configuration Editor for the project.
2. Select **Caching**, then **Result Caches**, then **Storage**.
3. In the **Cache file directory** box, type:
   `.\Caches\ServerDefinition`

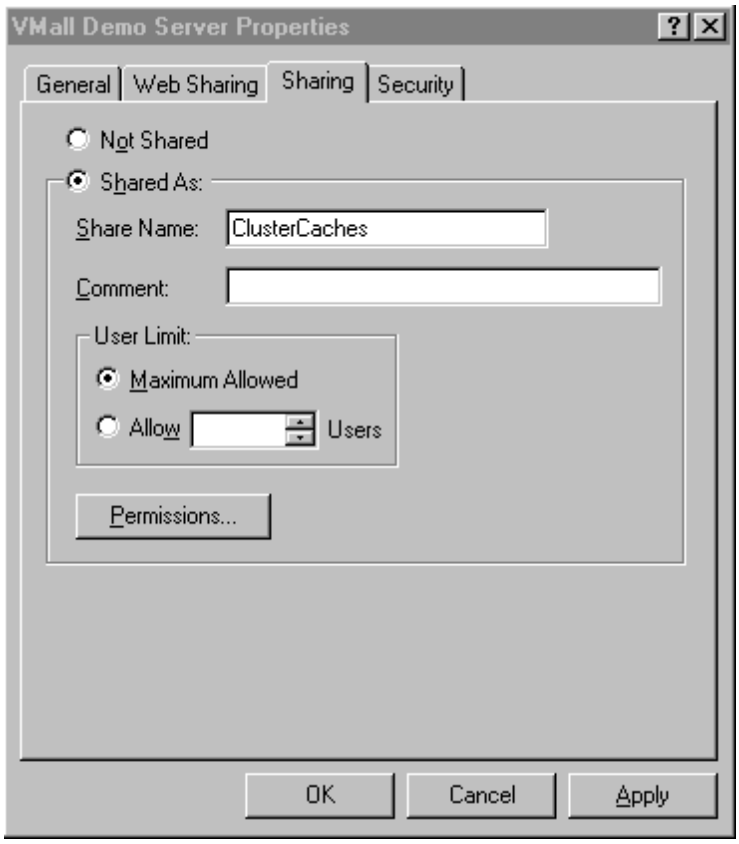
   where `ServerDefinition` is the name of the server definition.

   This tells the other clustered nodes to search for caches in the following path on all machines in the cluster:

   `<Intelligence Server Application Folder>\Caches\ServerDefinition`
4. Click **OK.**
5. On each machine in the cluster, open Windows Explorer and navigate to the cache file folder. The default location is:

   `C:\Program Files (x86)\Common Files\MicroStrategy\Intelligence Server\Caches\ServerDefinition`

   where `ServerDefinition` is the name of the server definition.
6. Right-click the cache file folder, and select **Sharing**. The [*Server Definition*] Properties dialog box opens.



7 On the Sharing tab, select the **Shared as** option. In the **Share Name** box, delete the existing text and type `ClusterCaches`.

8 Click **OK**. After you have completed these steps, you can cluster the nodes using the Cluster Monitor.

To configure cache sharing using a centralized cache file

1 Open the Project Configuration Editor for the project.

2 Select **Caching**, then **Result Caches**, then **Storage**.

3 In the **Cache file directory** box, type one of the following:

`\\<Machine Name>\<Shared Folder Name>`

or

`\\<IP Address>\<Shared Folder Name>`

For example, `\\My_File_Server\My_Cache_Directory`.

4 Click **OK**.

 © 2016, MicroStrategy Inc.

5 On the machine that is storing the centralized cache, create the file folder that will be used as the shared folder. The file folder name must be identical to the name you earlier specified in the **Cache file directory** box (shown as *Shared Folder Name* above).

Make sure this cache directory is writable to the network account under which Intelligence Server is running. Each Intelligence Server creates its own subdirectory.

Configuring History List sharing using multiple local cache files

If you are using a file-based History List, you can set up History Lists to use multiple local disk backups on each node in the cluster, using a procedure similar to the procedure above, *To configure cache sharing using multiple local cache files, page 381*. The History List messages are stored in the History folder. (To locate this folder, in the Intelligence Server Configuration Editor, expand **Governing Rules**, expand **Default**, then select **History settings**.)

The History List location is `.\Inbox\ServerDefinition`, where *`ServerDefinition`* is the name of the folder containing the History Lists. This folder must be shared with the share name "ClusterInbox" because this is the share name used by Intelligence Server to look for History Lists on other nodes.

Configuring caches in a cluster on UNIX/Linux

To configure a cluster of Intelligence Servers in a UNIX/Linux environment, all servers must have access to each others' caches and inbox (History List) files. Both cache and History List files are referred to generally as cache files throughout this section. An Intelligence Server looks for cache files from other nodes in the cluster by machine name. For an explanation and diagrams of general cache synchronization setup, see *Synchronizing cached information across nodes in a cluster, page 373*.

The cache and Inbox folders must be named as follows:

```
/<machine_name>/ClusterCaches

/<machine_name>/ClusterInbox
```

For example, a two-node cluster with Intelligence Servers is running on machines UNIX1 and UNIX2. Intelligence Server running on UNIX1 looks for caches of the other Intelligence Server only on `/UNIX2/ClusterCaches`.

The procedures below demonstrate how to configure the caches on two servers, named UNIX1 and UNIX2. Use these steps as a guideline for configuring your own system.

You can choose to use either procedure below, depending on whether you want to use centralized or local caching. For a detailed description and diagrams of cache synchronization setup, see *Synchronizing cached information across nodes in a cluster, page 373*.

© 2016, MicroStrategy Inc.

To configure a cluster with a centralized cache

This procedure assumes that the UNIX/Linux machines are called UNIX1 and UNIX2.

To configure the server definition and project

1. Start Intelligence Server on UNIX1.
2. In Developer, create project sources pointing to UNIX1 and UNIX2.
3. Connect to UNIX1 using Developer.
4. Right-click the project source of UNIX1 and select **Configure Server**.
5. Select the **Server Definition** category, and select **History Settings**.
6. Set the path using the following convention:

   `//<SharedLocation>/<InboxFolder>`

   In this example, set it as `//sandbox/Inbox`.
7. Right-click the project name and select **Project Configuration**.
8. Select **Caching**, then **Result Caches**, then **Storage**.
9. Following the convention `//<SharedLocation>/<CacheFolder>`, set the path to `//sandbox/Caches`.

   For caches stored on UNIX/Linux machines using Samba, set the path to `\\<machine name>\<shared folder name>`.
10. Disconnect from the project source and shut down Intelligence Server.

To create the cache folder on the shared device

11. Create the folders for caches on the shared device (as described in *Prerequisites for UNIX/Linux clustering, page 379* above):

    ```
    mkdir /sandbox/Caches
    ```

    ```
    mkdir /sandbox/Inbox
    ```
12. Restart both Intelligence Servers.

To configure a cluster with multiple local cache files

This procedure makes the following assumptions:

 © 2016, MicroStrategy Inc.

- The UNIX/Linux machines are called UNIX1 and UNIX2.
- Intelligence Server is installed in `MSTR_<HOME_PATH>` on each machine.
- The `MSTR_HOME_PATH` for each machine is `/Build/BIN/SunOS/`.

To configure the server definition and project

1. Start Intelligence Server on UNIX 1.
2. In Developer, create project sources pointing to UNIX1 and UNIX2.
3. Connect to UNIX1 using Developer.
4. Right-click the project source of UNIX1 and select **Configure Server**.
5. Select the **Server Definition** category, and select **History Settings**.
6. Set the path to `./ClusterInbox` and click **OK**.
7. Right-click the project name and select **Project Configuration**.
8. Select **Caching**, then **Result Caches**, then **Storage**.
9. Set the path for the cache file directory to `./ClusterCaches`.
10. Disconnect from the project source and shut down Intelligence Server.

To set up the UNIX1 machine

11. Create the folders for caches:

```
mkdir $MSTR_<HOME_PATH>/ClusterCaches
mkdir $MSTR_<HOME_PATH>/ClusterInbox
```

12. Mount the folders from UNIX2 on UNIX1. For example:

```
mkdir /UNIX2
mount UNIX2:/Build/BIN/SunOD /UNIX2
```

To set up the UNIX2 machine

13. Create the folders for caches:

```
mkdir $MSTR_HOME_PATH/ClusterCaches
mkdir $MSTR_HOME_PATH/ClusterInbox
```

14. Mount the folders from UNIX2 on UNIX1. For example:

```
mkdir /UNIX1
mount UNIX1:/Build/BIN/SunOD /UNIX1
```

15. Restart both Intelligence Servers.

© 2016, MicroStrategy Inc.

Configuring History Lists in a clustered environment

MicroStrategy recommends that you enable user affinity clustering to reduce History List resource usage. User affinity clustering causes Intelligence Server to connect all sessions for a user to the same node of the cluster. For background information about user affinity clustering, see *Synchronizing History Lists, page 377*.

If you are not using user affinity clustering, MicroStrategy recommends that you set the cache backup frequency to 0 (zero) to ensure that History List messages are synchronized correctly between nodes. For more information about this setting, see *Backup Frequency (minutes), page 417*.

To configure the History List governing settings for a clustered environment

1. In Developer, log into a project source. You must log in as a user that has administrative privileges.
2. From the **Administration** menu, point to **Server** and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **Server Definition** category, and then select **Advanced**.
4. Do one of the following:
   - To enable user affinity clustering, select the **User Affinity Cluster** check box.
   - OR, if you do not want to enable user affinity clustering, in the **Backup frequency (minutes)** field, type **0** (zero).
5. Click **OK** to accept your changes and close the Intelligence Server Configuration Editor.
6. Restart Intelligence Server.

Joining the nodes in a cluster

You join one node (or machine) to another node to form a cluster using the Cluster Monitor.

To join a node to a cluster

1. In Developer, log in to a project source. You must log in as a user with the Administer Cluster privilege.

 © 2016, MicroStrategy Inc.

2. Expand **Administration**, then expand **System Administration**, and then select **Cluster**. Information about each node in the cluster information displays on the right-hand side.
3. From the **Administration** menu, point to **Server**, then select **Join cluster**. The Cluster Manager dialog box opens.
4. Type the name of the machine running Intelligence Server to which you are adding this node, or click **...** to browse for and select it.
5. Once you have specified or selected the server to join, click **OK**.

Verifying the clustered system is working

Once all nodes have been synchronized and added to the cluster, you can verify that the cluster is working properly.

To verify from Developer

Verify the Cluster view

1. Connect to one Intelligence Server in the cluster and ensure that the Cluster view in Developer (under Administration, under System Administration) is showing all the proper nodes as members of the cluster.

Verify the cache

2. Connect to any node and run a large report.
3. Use the Cache Manager and view the report details to make sure the cache is created.
4. Connect to a different node and run the same report. Verify that the report used the cache created by the first node.

Verify the History List

5. Connect to any node and run a report.
6. Add the report to the History List.
7. Without logging out that user, log on to a different node with the same user name.
8. Verify that the History List contains the report added in the first node.

To verify from MicroStrategy Web

1. Open the MicroStrategy Web Administrator page.

2 Connect to any node in the cluster. MicroStrategy Web Universal should automatically recognize all nodes in the cluster and show them as connected.

If MicroStrategy Web does not recognize all nodes in the cluster, it is possible that the machine itself cannot resolve the name of that node. MicroStrategy cluster implementation uses the names of the machines for internal communication. Therefore, the Web machine should be able to resolve names to IP addresses. You can edit the `lmhost` file to relate IP addresses to machine names.

You can also perform the same cache and History List tests described above in *To verify from Developer, page 387*.

Distributing projects across nodes in a cluster

You can distribute projects across nodes of a cluster in any clustered configuration. Each node can host a different set of projects, which means only a subset of projects needs to be loaded on an Intelligence Server. This provides you with flexibility in using your resources and better scalability and performance.

To distribute projects across the cluster, you manually assign the projects to specific nodes in the cluster. Once a project has been assigned to a node, it is available for use.

If you do not assign a project to a node, the project remains unloaded and users cannot use it. You must then manually load the project for it to be available. To manually load a project, right-click the project in the Project Monitor and select **Load**.

If you are using single instance session logging in Enterprise Manager with clustered Intelligence Servers, the single instance session logging project must be loaded onto all the clustered Intelligence Servers. Failure to load this project on all servers at startup results in a loss of session statistics for any Intelligence Server onto which the project is not loaded at startup. For more information, see MicroStrategy Tech Note TN14591. For detailed information about session logging in Enterprise Manager, see the *MicroStrategy Operations Manager Guide*.

To distribute projects across nodes in a cluster

1 In Developer, from the **Administration** menu, point to **Projects**, then select **Select Projects**. Intelligence Server Configuration Editor opens, at the **Projects: General** category.

2 One column is displayed for each node in the cluster that is detected at the time the Intelligence Server Configuration Editor opens. Select the corresponding check box to configure the system to load a project on a node. A selected box at the intersection of a project row and a node column signifies that the project is to be loaded at startup on that node.

 © 2016, MicroStrategy Inc.

- If no check boxes are selected for a project, the project is not loaded on any node at startup. Likewise, if no check boxes are selected for a node, no projects are loaded on that node at startup.

If you are using single instance session logging with Enterprise Manager, the single instance session logging project must be loaded onto all the clustered Intelligence Servers at startup. Failure to load this project on all servers at startup results in a loss of session statistics for any Intelligence Server onto which the project is not loaded at startup. For steps on implementing single instance session logging, see the *MicroStrategy Operations Manager Guide*. For more information about this issue, see MicroStrategy Tech Note TN14591.

- **All Servers**: If this check box is selected for a project, all nodes in the cluster load this project at startup. All individual node check boxes are also selected automatically. When you add a new node to the cluster, any projects set to load on All Servers automatically load on the new node.

If you select a check mark for a project to be loaded on every node but you do not select the **All Servers** check box, the system loads the project on the selected nodes. When a new node is added to the cluster, this project is not automatically loaded on that new node.

3. Select whether to display only the selected projects and whether to apply the startup configuration on save:
   - **Show selected projects only**: Selecting this option allows you to display only those projects that have been assigned to be loaded on a node. For display purposes it filters out projects that are not loaded on any node in the cluster.
   - **Apply startup configuration on save**: Selecting this option allows your changes to be reflected immediately across the cluster. If this check box is cleared, any changes are saved when you click **OK**, but they do not take effect until Intelligence Server is restarted.
4. Click **OK** when you are finished configuring your projects across the nodes in the cluster.

If you do not see the projects you want to load displayed in the Intelligence Server Configuration Editor, you must configure Intelligence Server to use a server definition that points to the metadata containing the project. Use the MicroStrategy Configuration Wizard to configure this. For details, see the Installation and Configuration Guide.

It is possible that not all projects in the metadata are registered and listed in the server definition when the Intelligence Server Configuration Editor opens. This can occur if a project is created or duplicated in a two-tier (direct connection) project source that points to the same metadata as that being used by Intelligence Server while it is running. Creating, duplicating, or deleting a project in two-tier while a server is started against the same metadata is not recommended.

© 2016, MicroStrategy Inc.

Managing your clustered system

Once your clustered system is up and running, you can monitor and configure the projects that are running on each node of the cluster.

- *Managing your projects across nodes of a cluster, page 390*
- *Project failover and latency, page 392*
- *Shutting down a node, page 394*
- *Maintaining result caches and History Lists in a clustered environment, page 395*

Managing your projects across nodes of a cluster

Managing a project across all nodes of a cluster can be done through the Project view of the System Administration monitor. From this view, you can unload or idle a project during System Administration on Intelligence Server. However, sometimes you need to perform maintenance on only one node of the cluster. In this case, you can use the Cluster view to idle or unload a project from that node, while leaving the project running on the other nodes of the cluster.

For detailed information about the effects of the various idle states on a project, see *Setting the status of a project, page 27*.

To manage the projects and nodes in a cluster

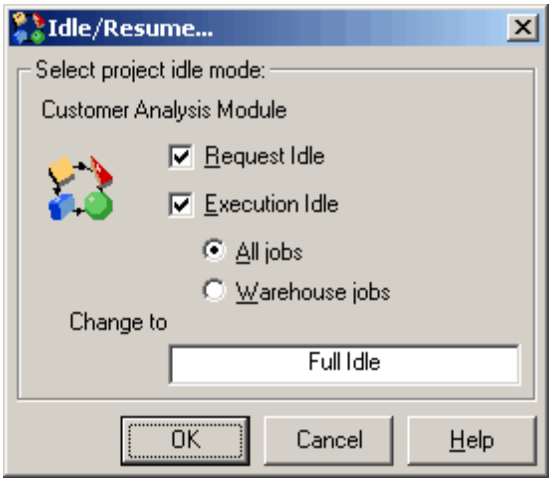
1. In Developer, log in to a project source. You must log in as a user with the Administer Cluster privilege.
2. Expand **Administration**, then expand **System Administration**, and then select **Cluster**. Information about each node in the cluster information displays on the right-hand side.
3. To see a list of all the projects on a node, click the **+** sign next to that node. The status of the project on the selected server is shown next to the project's name.

You can perform an action on multiple servers or projects at the same time. To do this, select several projects (**CTRL**+click), then right-click and select one of the options.

 © 2016, MicroStrategy Inc.

To idle or resume a project on a node

1 In the Cluster view, right-click the project whose status you want to change, point to **Administer project on node**, and select **Idle/Resume**. The Idle/Resume dialog box opens.



2 Select the options for the idle mode that you want to set the project to:

- Request Idle (**Request Idle**): all executing and queued jobs finish executing, and any newly submitted jobs are rejected.
- Execution Idle (**Execution Idle** for **All Jobs**): all executing, queued, and newly submitted jobs are placed in the queue, to be executed when the project resumes.
- Warehouse Execution Idle (**Execution Idle** for **Warehouse jobs**): all executing, queued, and newly submitted jobs that require SQL to be submitted to the data warehouse are placed in the queue, to be executed when the project resumes. Any jobs that do not require SQL to be executed against the data warehouse are executed.
- Full Idle (**Request Idle** and **Execution Idle** for **All jobs**): all executing and queued jobs are canceled, and any newly submitted jobs are rejected.
- Partial Idle (**Request Idle** and **Execution Idle** for **Warehouse jobs**): all executing and queued jobs that do not submit SQL against the data warehouse are canceled, and any newly submitted jobs are rejected. Any executing and queued jobs that do not require SQL to be executed against the data warehouse are executed.

To resume the project from a previously idled state, clear the **Request Idle** and **Execution Idle** check boxes.

3 Click **OK**. The Idle/Resume dialog box closes and the project goes into the selected mode.

To load or unload a project from a specific node

In the Cluster view, right-click the project whose status you want to change, point to **Administer project on node**, and select **Load** or **Unload**. The project is loaded or unloaded from that node.

Project failover and latency

Project failover support in a cluster is similar to system failover support. For example, one server in a cluster is hosting project A and another server in the cluster is running projects B and C. If the first server becomes unavailable, the other can begin running all three projects. Project failover support ensures that projects remain available even if hardware or an application fails.

Project failover is triggered when the number of nodes running a project reaches zero due to node failure. At that point, the system automatically loads any projects that were on the failed system onto another server in the cluster to maintain the availability of those projects. Once the failed server recovers, the system reloads the original project onto the recovered server. It also removes the project from the server that had temporarily taken over.

Failover and latency take effect only when a server fails. If a server is manually shut down, its projects are not automatically transferred to another server, and are not automatically transferred back to that server when it restarts.

You can determine several settings that control the time delay, or latency period, in the following instances:

- After a machine fails, but before its projects are loaded onto to a different machine
- After the failed machine is recovered, but before its original projects are reloaded

To set project failover latency

1. In Developer, from the **Administration** menu, select **Server**, then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
2. Expand the **Server Definition** category, then select **Advanced**.
3. Enter the **Project Failover Latency** and **Configuration Recovery Latency**, and click **OK**.

When deciding on these latency period settings, consider how long it takes an average project in your environment to load on a machine. If your projects are large, they may take some time to load, which presents a strain on your system resources. With this consideration in mind, use the following information to decide on a latency period.

 © 2016, MicroStrategy Inc.

Project failover latency

You can control the time delay (latency) before the project on a failed machine is loaded on another node to maintain a minimum level of availability.

Latency takes effect only when a server fails. If a server is manually shut down, its projects are not automatically transferred to another machine.

Consider the following information when setting a latency period:

- Setting a higher latency period prevents projects on the failed server from being loaded onto other servers quickly. This can be a good idea if your projects are large and you trust that your failed server will recover quickly. A high latency period provides the failed server more time to come back online before its projects need to be loaded on another server.
- Setting a lower latency period causes projects from the failed machine to be loaded relatively quickly onto another server. This is good if it is crucial that your projects are available to users at all times.
- Disabling the latency period or the failover process:
    - If you enter 0 (zero), there is no latency period and thus there is no delay; the project failover process begins immediately.
    - If you enter -1, the failover process is disabled and projects are not transferred to another node if there is a machine failure.

Configuration recovery latency

When the conditions that caused the project failover disappear, the system automatically reverts to the original project distribution configuration by removing the project from the surrogate server and loading the project back onto the recovered server (the project's original server).

Consider the following information when setting a latency period:

- Setting a higher latency period leaves projects on the surrogate server longer. This is good idea if your projects are large and you want to be sure your recovered server stays online for a specific period before the project load process begins. A high latency period provides the recovered server more time after it comes back online before its projects are reloaded.
- Setting a lower latency period causes projects on the surrogate machine to be removed and loaded relatively quickly onto the recovered server. This is desirable if you want to reduce the strain on the surrogate server as soon as possible.

You can also disable the latency period:

- If you enter a 0 (zero), there is no latency period and thus there is no delay. The configuration recovery process begins immediately.
- If you enter a -1, the configuration recovery process is disabled and projects are never automatically reloaded onto the recovered server.

© 2016, MicroStrategy Inc.

Shutting down a node

A node can be shut down in two ways:

- **Administrative shutdown**: This includes instances when a node is removed from a cluster or the Intelligence Server service is stopped.
- **Node failure**: This includes instances such as a power failure or a software error; this is sometimes called a forceful shutdown. Forcefully shutdown nodes retain their valid caches if they are available. However, while the node is shut down, there is no way to monitor the caches, change their status, or invalidate them. They can be deleted by manually deleting the cache files on the local node or by deleting the appropriate cache files on a shared network location. Be aware that cache files are named with object IDs.

The results of each of these types of shutdown are discussed below.

Resource availability

If a node is rendered unavailable because of a forceful shutdown, its cache resources are still valid to other nodes in the cluster and are accessed if they are available. If they are not available, new caches are created on other nodes.

In an administrative shutdown, caches associated with the shut down node are no longer valid for other nodes, even if they are physically available, such as on a file server.

Client connection status

Developer

Client connections that are not cluster-aware, such as Developer, do not experience any change if a node is removed from a cluster. However, the local node must regenerate its own caches rather than accessing the resources of other nodes. If Intelligence Server is shut down, any Developer clients connected to that Intelligence Server receive an error message notifying them of the lost connection, regardless of whether that Intelligence Server was in a cluster.

MicroStrategy Web

If a cluster node shuts down while MicroStrategy Web users are connected, those jobs return an error message by default. The error message offers the option to resubmit the job, in which case MicroStrategy Web automatically reconnects the user to another node.

Customizations to MicroStrategy Web can alter this default behavior in several ways.

If a node is removed from the cluster, all existing connections continue to function and remain connected to that machine, although the machine no longer has access to the

 © 2016, MicroStrategy Inc.

clustered nodes' resources. Future connections from MicroStrategy Web will be to valid cluster nodes.

Status after reboot

If a node goes down for any reason, all jobs on that node are terminated. Restarting the node provides an empty list of jobs in the job queue.

If a node is forcefully shut down in a Windows environment, it automatically rejoins the cluster when it comes back up.

Note the following:

- You can define the nodes that should automatically rejoin the cluster on restart from the Intelligence Server Configuration Editor. For steps on configuring this, see the *MicroStrategy Developer Help*.
- If multiple nodes in the cluster are restarted at the same time, they may not all correctly rejoin the cluster. To prevent this, separate the restart times by several minutes.

The nodes that are still in the cluster but not available are listed in the Cluster Monitor with a status of Stopped.

Maintaining result caches and History Lists in a clustered environment

Proper maintenance of result caches and History Lists is important in any MicroStrategy system. For detailed information on caches and cache management, including recommended best practices, see *Result caches, page 400*. For detailed information on History Lists, including best practices, see *Saving report results: History List, page 425*.

When maintaining result caches and History Lists in a clustered environment, be aware of the following:

- You can manage the caches on a node only if that node is active and joined to the cluster and if the project containing the caches is loaded on that node.
- Whenever a cache on one node of the cluster is created or updated, any copies of the old cache for that report, on the same node or on other nodes, are automatically invalidated. This means that only one valid copy of a cache exists at any time for a report on all nodes in the cluster. For more information about invalidating caches, see *Invalidating result caches, page 413*.
- The Cache Monitor's hit count number on a machine reflects only the number of cache hits that machine initiated on any cache in the cluster. If a different machine in the cluster hits a cache on the local machine, that hit is not be counted on the local machine's hit count. For more information about the Cache Monitor, see *Monitoring result caches, page 409*.

© 2016, MicroStrategy Inc.

For example, ServerA and ServerB are clustered, and the cluster is configured to use local caching (see *Local caching, page 375*). A report is executed on ServerA, creating a cache there. When the report is executed on ServerB, it hits the report cache on ServerA. The cache monitor on ServerA does not record this cache hit, because ServerA's cache monitor displays activity initiated by ServerA only.

- To ensure that History List messages are synchronized correctly between nodes and to reduce system overhead, either enable user affinity clustering or set the cache backup frequency to 0 (zero). For a discussion of these settings, including instructions, see *Configuring History Lists in a clustered environment, page 386*.

Maintaining History Lists in a clustered environment

User affinity clustering causes Intelligence Server to connect all sessions for a user to the same node of the cluster. This enables Intelligence Server to keep the user's History List on one node of the cluster. Resource use is minimized because the History List is not stored on multiple machines, and the History List is never out of sync across multiple nodes of the cluster.

MicroStrategy recommends that you enable user affinity clustering in any clustered system. If you are not using user affinity clustering, MicroStrategy recommends that you set the cache backup frequency to 0 (zero) to ensure that History List messages are synchronized correctly among nodes. For more information about this setting, see *Backup Frequency (minutes), page 417*.

To configure the History List governing settings for a clustered environment

1. In Developer, log into a project source. You must log in as a user with administrative privileges.
2. From the **Administration** menu, point to **Server** and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **Server Definition** category, and then select **Advanced**.
4. Do one of the following:
   - To enable user affinity clustering, select the **User Affinity Cluster** check box.
   - OR, if you do not want to enable user affinity clustering, then in the **Backup frequency (minutes)** field, type **0** (zero).
5. Click **OK** to close the Intelligence Server Configuration Editor.
6. Restart Intelligence Server.

© 2016, MicroStrategy Inc.

Connecting MicroStrategy Web to a cluster

You connect MicroStrategy Web to a cluster using MicroStrategy Web's Administration page. If the Intelligence Servers are on the same subnet as MicroStrategy Web and are accessible by User Datagram Protocol (UDP), the MicroStrategy Web Administration page can dynamically list the servers by looking for the listener service running on the machines. If the server is listed that you want to connect to, you can connect from this page. Alternatively, you can type the server name.

If the machine selected is part of a cluster, the entire cluster appears on the Administration page and is labeled as a single cluster. Click **Help** on MicroStrategy Web's Administration page for steps to connect to an Intelligence Server.

Once MicroStrategy Web is connected to a cluster, all nodes reference the same project. Load balancing directs new Web connections to the least loaded node, as measured by user connections. Once connected to a node, the Web user runs all MicroStrategy activity on the same node.

If nodes are manually removed from the cluster, projects are treated as separate in MicroStrategy Web, and the node connected to depends on which project is selected. However, all projects are still accessing the same metadata.

Clustering and firewalls

Connecting to Intelligence Server from MicroStrategy Web through a firewall is the same process regardless of the cluster state. The only difference is that allowable ports, sources, and destinations may be available between MicroStrategy Web and each of the nodes in the cluster.

Exporting to PDF or Excel

In MicroStrategy Web, users can export reports to PDF or to Excel for later viewing. Users must have the Write privilege for the **Inbox** folder on the Intelligence Server machine to be able to export reports.

To export to PDF or Excel in a clustered environment, users must have the Write privilege for the **ClusterInbox** folder on all Intelligence Servers in the cluster. For instructions on how to set up the **ClusterInbox** folder, see *Configuring caches in a cluster, page 380*.

Node failure

MicroStrategy Web or Web Universal users can be automatically connected to another node when a node fails. To implement automatic load redistribution for these users, on the Web Administrator page, under **Web Server** select **Security**, and in the Login area select **Allow Automatic Login if Session is Lost**.

© 2016, MicroStrategy Inc.

 © 2016, MicroStrategy Inc.

10

IMPROVING REPORT AND DOCUMENT RESPONSE TIME: CACHING

Introduction

A *cache* is a result set that is stored on a system to improve response time in future requests. With caching, users can retrieve results from Intelligence Server rather than re-executing queries against a database.

Intelligence Server supports the following types of caches:

- **Result caches**: Report and document results that have already been calculated and processed, that are stored on the Intelligence Server machine so they can be retrieved more quickly than re-executing the request against the data warehouse. For more information on these, see *Result caches, page 400*.

  Intelligent Cubes can function in a similar fashion to result caches: they allow you to store data from the data warehouse in Intelligence Server memory, rather than in the database. Intelligent Cubes are part of the OLAP Services add-on to Intelligence Server. For detailed information about Intelligent Cubes, see the *MicroStrategy In-memory Analytics Guide*.

- The **History List** is a way of saving report results on a per-user basis. For more information, see *Saving report results: History List, page 425*.

© 2016, MicroStrategy Inc.

- **Element caches**: Most-recently used lookup table elements that are stored in memory on the Intelligence Server or Developer machines so they can be retrieved more quickly. For more information on these, see *Element caches, page 440*.
- **Object caches**: Most-recently used metadata objects that are stored in memory on the Intelligence Server and Developer machines so they can be retrieved more quickly. For more information on these, see *Object caches, page 451*.

You specify settings for all cache types except History List under Caching in the Project Configuration Editor. History List settings are specified in the Intelligence Server Configuration Editor.

Result, element, and object caches are created and stored for individual projects; they are not shared across projects. History Lists are created and stored for individual users.

To make changes to cache settings, you must have the Administer Caches privilege. In addition, changes to cache settings do not take effect until you stop and restart Intelligence Server.

For additional ways to improve your MicroStrategy system's response time, see *Chapter 8, Tuning Your System for Best Performance*.

Result caches

A *result cache* is a cache of an executed report or document that is stored on Intelligence Server. Result caches are either report caches or document caches.

You cannot create or use result caches in a direct (two-tier) environment. Caches are stored in Intelligence Server, not retained on Developer.

Report caches can be created or used for a project only if the **Enable report server caching** check box is selected in the Project Configuration Editor under the **Caching: Result Caches: Creation** category.

Document caches can be created or used for a project only if the **Enable Document Output Caching in Selected Formats** check box is selected in the Project Configuration Editor under the **Caching: Result Caches: Creation** category, and one or more formats are selected.

Document caches are created or used only when a document is executed in MicroStrategy Web. Document caches are not created or used when a document is executed from Developer.

By default, result caching is enabled at the project level. It can also be set per report and per document. For example, you can disable caching at the project level, and enable caching only for specific, frequently used reports. For more information, see *Result cache settings at the report level, page 424*.

A result cache is created when you do any of the following:

© 2016, MicroStrategy Inc.

- In MicroStrategy Web or Developer, execute a saved report or document containing only static objects.
- In MicroStrategy Web or Developer, execute a saved report or document containing one or more prompts. Each unique set of prompt selections corresponds to a distinct cache.
- In MicroStrategy Web, execute a template and filter combination.
- Execute a report or document based on a schedule. The schedule may be associated with MicroStrategy Web, Developer, Mobile, Distribution Services, or Narrowcast Server. For more information about scheduling reports, see *Scheduling reports and documents: Subscriptions, page 488*.

Caching does not apply to a drill report request because the report is constructed on the fly.

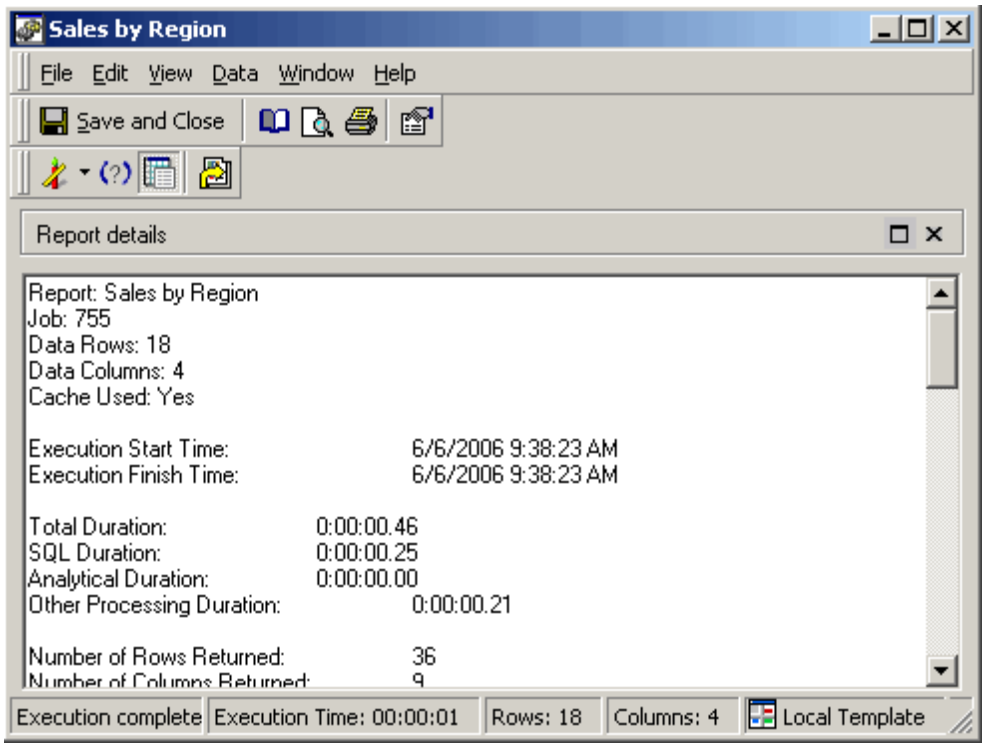
When a user runs a report (or, from MicroStrategy Web, a document), a job is submitted to Intelligence Server for processing. If a cache for that request is not found on the server, a query is submitted to the data warehouse for processing, and then the results of the report are cached. The next time someone runs the report or document, the results are returned immediately without having to wait for the database to process the query.

The Cache Monitor displays detailed information about caches on a machine; for more information see *Monitoring result caches, page 409*.

If you are running Intelligence Server on HP-UX v2, and you notice a slow response time when using the Cache Monitor, see *Cache Monitor and Intelligent Cube Monitor performance, page 767* for steps you can take to improve performance.

You can easily check whether an individual report hit a cache by viewing the report in SQL View. The image below shows the SQL View of a MicroStrategy Tutorial report, Sales by Region. The fifth line of the SQL View of this report shows “Cache Used: Yes.”

© 2016, MicroStrategy Inc.



Client-side analytical processing, such as ad hoc data sorting, pivoting, view filters, derived metrics, and so on, does not cause Intelligence Server to create a new cache.

This section discusses the following topics concerning result caching:

- *Cache management best practices, page 402*
- *Types of result caches, page 403*
- *Location of result caches, page 404*
- *Cache matching algorithm, page 406*
- *Disabling result caching, page 408*
- *Monitoring result caches, page 409*
- *Managing result caches, page 412*
- *Configuring result cache settings, page 416*

Cache management best practices

Good result cache management practices depend on a number of factors, such as the number of reports and documents in the project, the available disk space for caches, the amount of personalization in reports and documents, and whether you are using clustered Intelligence Servers.

MicroStrategy recommends the following best practices for cache management:

 © 2016, MicroStrategy Inc.

- The drive that holds the result caches should always have at least 10% of its capacity available.
- In a project with many reports, consider enabling caching on a report-by-report basis. Use MicroStrategy Enterprise Manager to determine which reports are used often and thus are good candidates for caching. For information about Enterprise Manager, see the Enterprise Manager chapter in the *MicroStrategy Operations Manager Guide*. For information about enabling caching per report, see *Result cache settings at the report level, page 424*.
- Disable caching for reports and documents with a high amount of personalization, such as prompt answers or security filters.

  To reuse results for reports and documents with a high amount of personalization, use MicroStrategy OLAP Services to create Intelligent Cubes. For more information about OLAP Services, see the *MicroStrategy In-memory Analytics Guide*.
- If results are cached by user ID (see *Create caches per user, page 420*), it may be better to disable caching and instead use the History List. For information about the History List, see *Saving report results: History List, page 425*.
- Be aware of the various ways in which you can tune the caching properties to improve your system's performance. For a list of these properties, and an explanation of each, see *Configuring result cache settings, page 416*.
- If you are using clustered Intelligence Servers, caching presents additional maintenance requirements. For information on maintaining caches in a clustered system, see *Maintaining result caches and History Lists in a clustered environment, page 395*.

Types of result caches

The following types of result caches are created by Intelligence Server:

- *Matching caches, page 403*
- *History caches, page 404*
- *Matching-History caches, page 404*
- *XML caches, page 404*

All document caches are Matching caches; documents do not generate History caches or XML caches. Intelligent Cube reports do not create Matching caches.

Matching caches

Matching caches are the results of reports and documents that are retained for later use by the same requests later on. In general, Matching caches are the type of result caches that are used most often by Intelligence Server.

When result caching is enabled, Intelligence Server determines for each request whether it can be served by an already existing Matching cache. If there is no match, it then runs the report or document on the database and creates a new Matching cache that can be

reused if the same request is submitted again. This caching process is managed by the system administrator and is transparent to general users who simply benefit from faster response times.

History caches

History caches are report results saved for future reference in the History List by a specific user. When a report is executed, an option is available to the user to send the report to the History List. Selecting this option creates a History cache to hold the results of that report and a message in the user's History List pointing to that History cache. The user can later reuse that report result set by accessing the corresponding message in the History List. It is possible for multiple History List messages, created by different users, to refer to the same History cache.

The main difference between Matching and History caches is that a Matching cache holds the results of a report or document and is accessed during execution; a History cache holds the data for a History List message and is accessed only when that History List message is retrieved.

For more information about History Lists, see *Saving report results: History List, page 425*.

Matching-History caches

A Matching-History cache is a Matching cache that is referenced by at least one History List message. It is a single cache composed of a Matching cache and a History cache. Properties associated with the Matching caches and History caches discussed above correspond to the two parts of the Matching-History caches.

XML caches

An *XML cache* is a report cache in XML format that is used for personalized drill paths. It is created when a report is executed from MicroStrategy Web, and is available for reuse in Web. It is possible for an XML cache to be created at the same time as its corresponding Matching cache. XML caches are automatically removed when the associated report or History cache is removed.

To disable XML caching, select the **Enable Web personalized drill paths** option in the **Project definition: Drilling** category in the Project Configuration Editor. Note that this may adversely affect Web performance. For more information about XML caching, see *ACLs and personalized drill paths in MicroStrategy Web, page 66*.

Location of result caches

Separate result caches are created for each project on an Intelligence Server. They are kept in memory and on disk. The server manages the swapping of these caches between memory and disk automatically. Caches are automatically unloaded, beginning with the least recently used cache, until the maximum memory governing limits are reached.

 © 2016, MicroStrategy Inc.

The amount of memory available to store result caches is limited by the Memory Storage settings. For information, see *Maximum RAM usage, page 421*.

Result cache files

By default, result cache files are stored in the directory where Intelligence Server is installed `\Caches\ServerDefinition\Machine Name\`. Report caches are stored in this folder; document caches are stored in the `\RWDCache\` subfolder of this folder.

Report cache file format

Report caches are stored on the disk in a binary file format. Each report cache has two parts:

- **Cache<cache ID>_Info.che** contains information about the cache, such as the user and prompt answers.
- **Cache<cache ID>.che** contains the actual data for the cache.

Report cache index files

Intelligence Server creates two types of index files to identify and locate report caches:

- **CachePool.idx** is an index file that contains a list of all Matching and History caches and pointers to the caches' locations.
- **CacheLkUp.idx** is a lookup table that contains the list of all Matching caches and their corresponding cache keys. Incoming report requests are matched to report cache keys in this table to determine whether a Matching cache can be used. This process is called cache matching (see *Cache matching algorithm, page 406*). This lookup table is always backed up to disk when Intelligence Server shuts down. Additional backups are based on the Backup frequency and the Lookup Cleanup Frequency settings (see *Result cache settings at the server level, page 416*).

Document cache file format

Document caches are stored on the disk in a binary file format. Each document cache has two parts:

- **<cache ID>_info.rwdc** contains information about the cache, such as the user and prompt answers.
- **<cache ID>.rwdc** contains the actual data for the cache.

Document cache index files

Intelligence Server creates two types of index files to identify and locate document caches:

- **RWDPool.idx** is an index file that contains a list of all Matching caches and pointers to the caches' locations.
- **RWDLkUp.idx** is a lookup table that contains the list of all Matching caches and their corresponding cache keys. Incoming document requests from Web are matched to document cache keys in this table to determine whether a Matching cache can be used. This process is called cache matching (see *Cache matching algorithm, page 406*). The lookup table is always backed up to disk when Intelligence Server shuts down. Additional backups are based on the Backup frequency and the Lookup Cleanup Frequency settings (see *Result cache settings at the server level, page 416*).

Cache matching algorithm

When a user requests a report, or a document from Web, cache keys are used to determine whether a cache can be used to satisfy the request. If the cache keys in the request match the ones in the result cache, the cached report or document results are used. The matching process takes several steps that involve a number of cache keys, and each step is explained in detail below. If at any step, the matching is not successful, then the cache is not used and the request executes against the data warehouse.

Step 1: Check the IDs

To check whether the requested report/document and the cached report/document are the same, Intelligence Server compares the **ID** and **Version ID** of the two. If they match, the process continues to Step 2.

Alternately, Intelligence Server checks the **Template ID**, **Template Version ID**, **Filter ID**, and **Filter Version ID** in the requested report/document against the ones in the cache. If all of them match, the process continues to Step 2.

If you are not using MicroStrategy OLAP Services, any modification to a report, even a simple formatting change or an Access Control List (ACL) modification, changes the Template Version ID and invalidates the report cache. With MicroStrategy OLAP Services, the cache is invalidated only if the contents of the Report Objects pane change. For more information about OLAP Services, see *Intelligent Cubes, page 355*.

Step 2: Check the personalization impact

If the report or document contains prompts, Intelligence Server checks the **prompt answers** selected for the report. Different prompt answers change the content of the report; therefore, the cache is not used if the prompt answers in the report request are not the same as the ones in the report cache. Each set of distinct prompt answers creates a distinct cache.

Step 3: Check the security impact

Intelligence Server makes sure that users with different security filters cannot access the same cache. Intelligence Server compares the **Security ID** and **Security Version ID** of all the security filters applied to the user in the request, including those inherited from the

 © 2016, MicroStrategy Inc.

groups to which he or she belongs, with the security profile of the user who originated the cache.

Step 4: Check the modification impact

Intelligence Server does not use a cache if an object in the report/document changes. To check this, Intelligence Server compares the **IDs** and **Version IDs** of all application objects used in the requested report/document with the ones used in the cached report/document. If any of these IDs are different, the existing cache is automatically invalidated.

Step 5: Check the data language

Intelligence Server makes sure a cache is not used if the user running the report is using a different language than the user who created the cache. Each different language creates a different cache.

Step 6: Check the database security impact (optional)

You may find it necessary to add optional criteria, listed below, to the cache matching process. These criteria are useful if database security view and connection mapping are used to ensure that users with different security profiles, who see different data from the data warehouse, cannot access the same cache. For information about connection mapping, see *Controlling access to the database: Connection mappings, page 74*).

- **User ID**: To match caches by the global unique identifier (GUID) of the user requesting the cache, in the Caching: Result Caches: Creation category in the Project Configuration Editor, select the **Create caches per user** check box.
- **Database login**: To match caches by the GUID of the database login assigned to the user via a connection mapping, in the Caching: Result Caches: Creation category in the Project Configuration Editor, select the **Create caches per database login** check box.

  This option is especially useful if database warehouse authentication is used. For more information, see *Implementing database warehouse authentication, page 103*.
- **Database connection**: To match caches by the GUID of the database connection assigned to the user via a connection mapping, in the Caching: Result Caches: Creation category in the Project Configuration Editor, select the **Create caches per database connection** check box.

Step 7: Check additional criteria for documents

Document caches have additional criteria that must match before a cache can be used:

- The **Export Option** (All or Current Page) and **Locale** of the document must match the cache.

- The selector and group-by options used in the document must match those used in the cache.
- The format of the document (PDF, Excel, HTML, or XML/Flash) must match the format of the cache.
- In Excel, the document and cache must both be either enabled or disabled for use in MicroStrategy Office.
- In XML/Flash, the mode of the document (View, Interactive, Editable, Flash) must match the mode of the cache.
- In XML/Flash, the Web preferences of the user executing the document must match the Web preferences of the user who created the cache.

Disabling result caching

By default, result caching is enabled in Intelligence Server. If the performance gain is marginal compared to the added overhead, you can disable report caching. You may wish to disable caching in the following situations:

- The data warehouse is updated more than once a day.
- Most reporting is ad hoc so caching provides little value.
- Reports are heavily prompted, and the answer selections to the prompts are different each time the reports are run.
- Few users share the same security filters when accessing the reports.

If you disable result caching for a project, you can set exceptions by enabling caching for specific reports or documents. For more information, see *Result cache settings at the report level, page 424*.

To disable result caching

1. Open the Project Configuration Editor for the project.
2. Expand **Caching**, expand **Result Caches**, then select **Creation**.
3. To disable report and document caching, clear the **Enable report server caching** check box.
4. To disable document caching but not report caching, leave the **Enable report server caching** check box selected and clear the **Enable document output caching in selected formats** check box.
5. Click **OK**. Caching is now disabled for this project.

 © 2016, MicroStrategy Inc.

Monitoring result caches

You use the Cache Monitor in Developer to monitor result caches. When result caching is enabled and a user executes a report or document, a cache entry is listed in the Cache Monitor.

You can also use the Diagnostics Configuration Tool for diagnostic tracing of result caches (see *Diagnostics and Performance Logging Tool, page 411*), and Command Manager to automatically update information about result caches (see *Command Manager, page 412*).

A cache's hit count is the number of times the cache is used. When a report is executed (which creates a job) and the results of that report are retrieved from a cache instead of from the data warehouse, Intelligence Server increments the cache's hit count. This can happen when a user runs a report or when the report is run on a schedule for the user. This does not include the case of a user retrieving a report from the History List (which does not create a job). Even if that report is cached, it does not increase its hit count.

To view all report or document caches for a project in the Cache Monitor

1. In Developer, log in to a project source. You must log in as a user with the Monitor Caches privilege.
2. Expand **Administration**, then expand **System Monitors**, then expand **Caches**, and then select **Reports** or **Documents**. The Cache Monitor Options dialog box opens.
3. Select the project for which you want to view the caches and click **OK**. The Report Cache Monitor or Document Cache Monitor opens.
4. To view additional details about a cache, double-click that cache. A Cache Quick View dialog box opens for that cache.
5. To view additional details about all caches, from the **View** menu select **Details**.
6. To change the columns shown in the Details view, right-click in the Cache Monitor and select **View Options**. The Cache Monitor View Options dialog box opens. Select the columns you want to see and click **OK**.
7. To view caches from a different project, right-click in the Cache Monitor and select **Filter**. The Cache Monitor Options dialog box opens. Select the project for which you want to view caches and click **OK**. That project's caches are now shown in the Cache Monitor.
8. To display History and XML caches in the Report Cache Monitor, right-click in the Cache Monitor and select **Filter**. The Cache Monitor Options dialog box opens. Select **Show caches for History List messages** or **Show XML caches** and click **OK**. The XML or History caches for the specified project are now shown in the Report Cache Monitor.

© 2016, MicroStrategy Inc.

You can perform any of the following options after you select one or more caches and right-click:

- **Delete**: Removes the cache from both memory and disk
- **Invalidate**: Marks the cache as unusable, but leaves a reference to it in users' History Lists (if any)
- **Load from disk**: Loads into memory a cache that was previously unloaded to disk
- **Unload to disk**: Removes the cache from memory and stores it on disk

For detailed information about these actions, see *Managing result caches, page 412*.

If you are running Intelligence Server on HP-UX v2, you may notice a slow response time when using the Cache Monitor. For information about this delay, including steps you can take to improve performance, see *Cache Monitor and Intelligent Cube Monitor performance, page 767*.

Cache statuses

A result cache's status is displayed in the Report Cache Monitor using one or more of the following letters:

Status	Stands for	Description
R		The cache is valid and ready to be used.
P		The cache is currently being updated.
I		The cache has been invalidated, either manually or by a change to one of the objects used in the cache. It is no longer used, and will be deleted by Intelligence Server. For information about invalid caches, see *Invalidating result caches, page 413*.
E		The cache has been invalidated because its lifetime has elapsed. For information about expired caches, see *Expiring result caches, page 416*.
L		The cache is loaded into Intelligence Server memory.
U		The cache file has been updated.
D		The cache has been updated in Intelligence Server memory since the last time it was saved to disk.
F		The cache has been unloaded, and exists as a file on disk instead of in Intelligence Server memory. For information about loading and unloading caches, see *Unloading and loading result caches to disk, page 413*.

Cache types

Result caches can be of the following types:

 © 2016, MicroStrategy Inc.

Type	Description
Matching	The cache is valid and available for use. **Note**: All document caches are Matching caches.
History	The cache referenced in at least one History List message.
Matching-History	The cache is valid and available for use, and also referenced in at least one History List message.
XML	(Web only) The cache exists as an XML file and is referenced by the matching cache. When the corresponding Matching cache is deleted, the XML cache is deleted.

For more information about each type of cache, see *Types of result caches, page 403*.

Diagnostics and Performance Logging Tool

The Intelligence Server logs are often useful when troubleshooting issues with report caching in a MicroStrategy system. You can view these logs and configure what information is logged using the Diagnostics and Performance Logging Tool. For more information, see *Configuring what is logged, page 732*.

To enable diagnostic tracing of result caches

1. Open the MicroStrategy Diagnostics and Performance Logging Tool. (From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Diagnostics Configuration**.)
2. In the **Select Configuration** drop-down list, select **CastorServer Instance**.
3. Clear the **Use Machine Default Diagnostics Configuration** check box.
4. In the Report Server component, in the Cache Trace dispatcher, click the **File Log** (currently set to **<None>**) and select **<New>**. The Log Destination Editor opens.
5. Enter the following information in the editor:
   - **Select Log Destination**: <New>
   - **File Name**: cacheTrace
   - **Max File Size**: 5000
   - **File Type**: Diagnostics
6. Click **Save**, and then click **Close**. The Log Destination Editor closes.
7. In the Report Server component, in the Cache Trace dispatcher, click the **File Log** (currently set to **<None>**) and select **cacheTrace**. The creation and deletion of report caches is now logged to this file.

© 2016, MicroStrategy Inc.

Command Manager

You can also use the following Command Manager scripts to monitor result caches:

- `LIST [ALL] REPORT CACHES [FOR PROJECT "<project_name>"]` lists all report caches on Intelligence Server for a project.
- `LIST [ALL] PROPERTIES FOR REPORT CACHE "<cache_name>" IN PROJECT "<project_name>"` lists information about a report cache.

By default, these scripts are at `C:\Program Files (x86)\MicroStrategy\Command Manager\Outlines\Cache_Outlines`.

For more information about Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*, or the Command Manager Help (from within Command Manager, press **F1**).

Managing result caches

As a system administrator, your greatest concerns about caching are consistency and availability of the cached data. You have the important responsibility of synchronizing the caches with the data in the data warehouse. Therefore, as data changes in the data warehouse, you must ensure that the outdated cached data is either updated or discarded. You can do this in two main ways: Invalidating and Scheduling. These methods, along with other maintenance operations that you can use when managing result caches, are discussed below. They include:

- *Scheduling updates of result caches, page 412*
- *Unloading and loading result caches to disk, page 413*
- *Invalidating result caches, page 413*
- *Deleting result caches, page 414*
- *Purging all result caches in a project, page 415*
- *Expiring result caches, page 416*

Scheduling updates of result caches

You can schedule a report or document to be executed regularly, to ensure that the result cache is up-to-date. Scheduling is a proactive measure aimed at making sure result caches are readily available when needed.

Typically, reports and documents that are frequently used best qualify for scheduling. Reports and documents that are not frequently used do not necessarily need to be scheduled because the resource cost associated with creating a cache on a schedule might not be worth it. For more information on scheduling a result cache update, see *Scheduling reports and documents: Subscriptions, page 488*.

 © 2016, MicroStrategy Inc.

Unloading and loading result caches to disk

You may need to unload caches from memory to disk to create free memory for other operations on the Intelligence Server machine.

If a report cache is unloaded to disk and a user requests that report, the report is then loaded back into memory automatically. You can also manually load a report cache from the disk into memory.

Caches are saved to disk according to the Backup frequency setting (see *Backup Frequency (minutes), page 417*). Caches are always saved to disk regardless of whether they are loaded or unloaded; unloading or loading a cache affects only the cache's status in Intelligence Server memory.

Invalidating result caches

Invalidating a result cache indicates to Intelligence Server that this cache should not be used. Invalidation is a preventive measure that you can take to ensure that users do not run reports that are based on outdated cached data. Examples of when the data may be outdated include:

- When the data warehouse changes, the existing caches are no longer valid because the data may be out of date. In this case, future report/document requests should no longer use the caches.
- When the definition of an application object (such as a report definition, template, filter, and so on) changes, the related result cache is automatically marked as invalid.
- When the cache for any of the datasets for a document becomes invalidated or deleted, the document cache is automatically invalidated.

Caches need to be invalidated when new data is loaded from the data warehouse so that the outdated cache is not used to fulfill a request. You can invalidate all caches that rely on a specific table in the data warehouse. For example, you could invalidate all report/document caches that use the Sales_Trans table in your data warehouse.

Only Matching and Matching-History caches can be invalidated. Invalidating a cache has the following effects:

- An invalid Matching cache is automatically deleted.
- An invalid Matching-History cache is converted to a History cache. If all History messages relating to this cache are deleted, the converted History cache is also deleted.

MicroStrategy strongly recommends that you invalidate Matching and Matching-History caches instead of deleting them directly.

Invalid caches are deleted automatically based on the **Cache lookup cleanup** frequency setting. For more information about this setting, see *Cache lookup cleanup frequency (sec), page 417*.

You can invalidate caches manually or by scheduling the invalidation process.

© 2016, MicroStrategy Inc.

Invalidating a cache with a scheduled administration task

You can schedule a MicroStrategy administration task to invalidate caches on a recurring schedule. In the Project Configuration Editor, in the Caches: Result Caches (Maintenance) category, you can select a schedule to be used to invalidate caches. For more information about scheduling tasks, see *Scheduling administrative tasks, page 485*.

Invalidating a cache with a Command Manager script

You can update the data warehouse load routine to invoke a MicroStrategy Command Manager script to invalidate the appropriate caches. This script is at `C:\Program Files (x86)\MicroStrategy\Command Manager\Outlines\Cache_Outlines\Invalidate_Report_Cache_Outline`. For more information about Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*.

To invoke Command Manager from the database server, use one of the following commands:

- **SQL Server**: exec xp.cmdshell cmdmgr
- **Oracle**: host cmdmgr
- **DB2**: ! cmdmgr
- **Teradata**: os cmdmgr

Invalidating a cache manually

From the Cache Monitor, you can manually invalidate one or more caches.

To manually invalidate a cache

1. In Developer, log into a project source. You must log in as a user with the Monitor Caches privilege.
2. Expand **Administration**, then expand **System Monitors**, then expand **Caches**, and then select **Reports** or **Documents**. The Cache Monitor Options dialog box opens.
3. Select the project for which you want to invalidate a cache and click **OK**. The Report Cache Monitor or Document Cache Monitor opens.
4. Right-click the cache to invalidate and select **Invalidate Cache**. The cache is invalidated.

Deleting result caches

Typically, you do not need to manually delete result caches if you are invalidating caches and managing History List messages. Result caches are automatically deleted by

 © 2016, MicroStrategy Inc.

Intelligence Server if cache invalidation and History Lists are performed and maintained properly, as follows:

- A Matching cache is deleted automatically when it is invalidated.
- A History cache is deleted automatically when all History List messages that reference it are deleted. MicroStrategy recommends that you actively maintain History List messages, as History caches are deleted automatically.
- A Matching-History cache is handled in the following way:
  - When all the History List messages that reference a Matching-History cache are deleted, the cache is converted to a Matching cache.
  - When a Matching-History cache is invalidated, it is converted to a History cache.
- An XML cache is deleted automatically when its associated Matching or History cache is deleted.

In all cases, cache deletion occurs based on the Cache lookup cleanup frequency setting. For more information about this setting, see *Cache lookup cleanup frequency (sec), page 417*.

You can manually delete caches via the Cache Monitor and Command Manager, or schedule deletions via the Administration Tasks Scheduling, in the same way that you manually invalidate caches. For details, see *Invalidating result caches, page 413*.

Purging all result caches in a project

You can delete all the result caches in a project at once by selecting the **Purge Caches** option in the Project Configuration Editor. This forces reports executed after the purge to retrieve and display the latest data from the data warehouse.

Purging deletes all result caches in a project, including caches that are still referenced by the History List. Therefore, purge caches only when you are sure that you no longer need to maintain any of the caches in the project, and otherwise delete individual caches.

Even after purging caches, reports and documents may continue to display cached data. This can occur because results may be cached at the object and element levels, in addition to at the report/document level. To ensure that a re-executed report or document displays the most recent data, purge all three caches. For instructions on purging element and object caches, see *Deleting all element caches, page 450* and *Deleting object caches, page 454*.

To purge all result caches in a project

1. In Developer, right-click the project and select **Project Configuration Editor**. The Project Configuration Editor opens.
2. Expand **Caching**, then **Result Caches**, and then select **Maintenance**.

© 2016, MicroStrategy Inc.

3 Click **Purge Now**. The caches are purged.

Expiring result caches

Cache expiration is the process of marking a cache out of date. Expiring a cache has the same result as invalidating a cache, and applies to Matching caches and Matching-History caches. The only difference between expiration and invalidation is that expiration happens after a set period of time. For information on how invalidation works, see *Invalidating result caches, page 413*.

MicroStrategy strongly recommends that you invalidate a cache when changes in the data from the data warehouse affect the cache, rather than relying on a time interval to expire caches. To disable cache expiration, in the Caching: Result Caches: Maintenance subcategory of the Project Configuration Editor, select the **Never expire caches** check box.

Cache expiration occurs automatically according to the **Cache duration (Hours)** setting in the Caching: Result Caches (Maintenance) subcategory in the Project Configuration Editor.

When a cache is updated, the current cache lifetime is used to determine the cache expiration date based on the last update time of the cache. This means that changing the **Cache duration (Hours)** setting or the **Never Expire Caches** setting does not affect the expiration date of existing caches. It affects only the new caches that are being or will be created.

Configuring result cache settings

Result cache settings can be configured at three levels:

- At the server level (see *Result cache settings at the server level, page 416*)
- At the project level (see *Result cache settings at the project level, page 417*)
- At the individual report/document level (see *Result cache settings at the report level, page 424*)

Each is discussed in detail below.

Changes to any of the caching settings are in effect only after Intelligence Server restarts.

Result cache settings at the server level

You can configure the following caching settings in the Intelligence Server Configuration Editor, in the Server Definition (Advanced) category. Each is described below.

- *Backup Frequency (minutes), page 417*
- *Cache lookup cleanup frequency (sec), page 417*

 © 2016, MicroStrategy Inc.

You can also configure these settings using the Command Manager script, `Alter_Server_Config_Outline.otl`, located at `C:\Program Files (x86)\MicroStrategy\Command Manager\Outlines\Cache_Outlines`.

Backup Frequency (minutes)

When a result cache is created, the cache is initially stored in memory on Intelligence Server. Caches are backed up to disk as specified by the backup frequency setting.

You can specify the cache backup frequency in the **Backup frequency (minutes)** box under the **Server Definition: Advanced** subcategory in the Intelligence Server Configuration Editor.

If you specify a backup frequency of 0 (zero), result caches are saved to disk as soon as they are created. If you specify a backup frequency of 10 (minutes), the result caches are backed up from memory to disk ten minutes after they are created.

In a clustered environment, MicroStrategy recommends that you set the backup frequency to 0 (zero) to ensure that History List messages are synchronized correctly.

Backing up caches from memory to disk more frequently than necessary can drain resources.

This setting also defines when Intelligent Cubes are saved to secondary storage, as described in *Defining when Intelligent Cubes are automatically saved to secondary storage, page 473*.

Cache lookup cleanup frequency (sec)

The **Cache lookup cleanup frequency (sec)** setting determines how frequently the `CacheLkUp.idx` file is cleaned up. This file stores cache matching information and can become significant in size, especially when a large number of caches include a large number of prompts. The cleanup process reduces the amount of memory that the file consumes and the time that it takes to back up the lookup table to disk.

The default value for this setting is 0 (zero), which means that the cleanup takes place only at server shutdown. You may change this value to another based on your needs, but make sure that it does not negatively affect your system performance. MicroStrategy recommends cleaning the cache lookup at least daily but not more frequently than every half hour.

Result cache settings at the project level

You can configure the following caching settings in the Project Configuration Editor, in the Result Caches category. Each is described below.

- In the Result Caches (creation) subcategory
  - *Enable report server caching, page 418*
  - *Enable document output caching in selected formats, page 419*
  - *Enable prompted report and document caching, page 419*

    - *Record prompt answers for cache monitoring, page 419*
    - *Enable non-prompted report and document caching, page 419*
    - *Enable XML caching for reports, page 420*
    - *Create caches per user, page 420*
    - *Create caches per database login, page 420*
    - *Create caches per database connection, page 420*
- In the Result caches (storage) subcategory
    - *Cache file directory, page 420*
    - *Cache encryption level on disk, page 421*
    - *Maximum RAM usage, page 421* (separate settings for report and document caches)
    - *Maximum number of caches, page 422* (separate settings for report and document caches)
    - *RAM swap multiplier, page 422*
    - *Maximum RAM for cache index %, page 423*
    - *Load caches on startup, page 423*
- In the Result caches (maintenance) subcategory
    - *Never expire caches, page 423*
    - *Cache duration (Hours), page 423*
    - *Cache expiration and dynamic dates, page 424*
    - *Purging all result caches in a project, page 415*
- In the Subscription Execution subcategory
    - *Cache usage defaults for subscriptions, page 424*

To locate these settings, right-click the project and select **Project Configuration**. Then, in the Project Configuration Editor, expand **Caching**, and then select **Result Caches**.

You can also configure these settings using Command Manager scripts located at `C:\Program Files (x86)\MicroStrategy\Command Manager\Outlines\Cache_Outlines`.

Enable report server caching

Result caches can be created or used for a project only if the **Enable report server caching** check box is selected in the Project Configuration Editor in the **Caching: Result Caches: Creation** category.

 © 2016, MicroStrategy Inc.

If this option is disabled, all the other options in the Result Caches: Creation and Result Caches: Maintenance categories are grayed out, except for **Purge Now**. By default, report server caching is enabled. For more information on when report caching is used, see *Result caches, page 400*.

Enable document output caching in selected formats

Document caches can be created or used for a project only if the **Enable document output caching in selected formats** check box is selected in the Project Configuration Editor in the **Caching: Result Caches: Creation** category. Document caches are created for documents that are executed in the selected output formats. You can select all or any of the following: PDF, Excel, HTML, and XML/Flash/HTML5.

Document caches are created or used only when a document is executed from MicroStrategy Web. They are not created or used in Developer.

Enable prompted report and document caching

Enabled by default, the **Enable caching for prompted reports and documents** setting controls whether prompted reports and documents are cached. In an environment where the majority of reports are prompted and each prompt is likely to receive a different answer each time it is used, the probability of matching an existing cache is low. In this case, caching these report datasets do not provide significant benefits; therefore you may want to disable this setting.

To disable this setting, clear its check box in the Project Configuration Editor under the **Caching: Result Caches: Creation** category.

Record prompt answers for cache monitoring

If you **Enable caching for prompted reports and documents** (see above), you can also **Record prompt answers for cache monitoring**. This causes all prompt answers to be listed in the Cache Monitor when browsing the result caches. You can then invalidate specific caches based on prompt answers, either from the Cache Monitor or with a custom Command Manager script.

This option is disabled by default. To enable it, select its check box in the Project Configuration Editor under the Caching: Result Caches: Creation category.

Enable non-prompted report and document caching

If you **Enable caching for non-prompted reports and documents**, reports and documents without any prompts are cached.

This option is enabled by default. To disable it, clear its check box in the Project Configuration Editor under the **Caching: Result Caches: Creation** category.

© 2016, MicroStrategy Inc.

Enable XML caching for reports

If you **Enable XML caching for reports**, reports executed from MicroStrategy Web create XML caches in addition to any Matching or History caches they may create. For information about XML caches, see *XML caches, page 404*.

This option is enabled by default. To disable it, clear its check box in the Project Configuration Editor under the **Caching: Result Caches: Creation** category.

Create caches per user

If the **Create caches per user** setting is enabled, different users cannot share the same result cache. Enable this setting only in situations where security issues (such as database-level Security Views) require users to have their own cache files. For more information, see *Cache matching algorithm, page 406*.

Instead of enabling this setting, it may be more efficient to disable caching and instead use the History List. For information about the History List, see *Saving report results: History List, page 425*.

This option is disabled by default. To enable it, select its check box in the Project Configuration Editor under the **Caching: Result Caches: Creation** category.

Create caches per database login

Select the **Create caches per database login** option if database authentication is used. This means that users who execute their reports using different database login IDs cannot use the same cache. For more information, see *Cache matching algorithm, page 406*.

This option is disabled by default. To enable it, select its check box in the Project Configuration Editor under the **Caching: Result Caches: Creation** category.

Create caches per database connection

Select the **Create caches per database connection** option if connection mapping is used. For more information, see *Cache matching algorithm, page 406*.

This option is disabled by default. To enable it, select its check box in the Project Configuration Editor under the **Caching: Result Caches: Creation** category.

Cache file directory

The **Cache file directory**, in the Project Configuration Editor under the Caching: Result Caches: Storage category, specifies where all the cache-related files are stored. By default these files are stored in the Intelligence Server installation directory, in the `\Caches\<Server definition name>` subfolder.

In a non-clustered environment, report caches are typically stored on the same machine that is running Intelligence Server.

 © 2016, MicroStrategy Inc.

In a clustered environment, there are two options:

- **Local caching**: Each node hosts its own cache file directory that needs to be shared as "ClusterCache" so that other nodes can access it. ClusterCache is the share name Intelligence Server looks for on other nodes to retrieve caches.
- **Centralized caching**: All nodes have the cache file directory set to the same network location, `\\<machine name>\<shared directory name>`. For example, `\\My_File_Server\My_Cache_Directory`.

Note the following:

- For caches located on Windows machines, and on UNIX/Linux machines using Samba, set the path to `\\<machine name>\<shared directory name>`. For caches stored on UNIX/Linux machines, set the path to `//<SharedLocation>/<CacheFolder>`.
- On UNIX systems, it is recommended that you mount the shared location as a network drive. You must create a folder in your machine's `Volumes` directory before mounting the location. For example, `mount -t afp afp://my_file_server/my_inbox_directory /Volumes/my_network_mount`

Make sure this cache directory is writable from the network account under which Intelligence Server is running. Each Intelligence Server creates its own subdirectory.

For more information about which configuration may be best in clustered environments, see *Configuring caches in a cluster, page 380*.

Cache encryption level on disk

The **Cache encryption level on disk** drop-down list controls the strength of the encryption on result caches. You can configure result caches to use either simple encryption or AES encryption with a 128-bit key. Encrypting caches increases security, but may slow down the system.

By default the caches that are saved to disk are not encrypted. You can change the encryption level in the Project Configuration Editor under the **Caching: Result Caches: Storage** category.

Maximum RAM usage

The **Maximum RAM usage** settings, in the Project Configuration Editor under the **Caching: Result Caches: Storage** category, control the amount of memory that result caches consume on Intelligence Server. When this setting is about to be exceeded, the least recently used caches are automatically unloaded to disk.

If the machine experiences problems because of high memory use, you may want to reduce the **Maximum RAM usage** for the result caches. You need to find a good balance between allowing sufficient memory for report caches and freeing up memory for other uses on the machine. The default value is 25 megabytes for reports and datasets, and 256

© 2016, MicroStrategy Inc.

megabytes for formatted documents. The maximum value for each of these is 65536 megabytes, or 64 gigabytes.

MicroStrategy recommends that you initially set this value to 10% of the system RAM if it is a dedicated Intelligence Server machine, that is, if no other processes are running on it. This setting depends on the following factors:

- The size of the largest report cache.

  This setting should be at least as large as the largest report in the project that you wish to be cached. If the amount of RAM available is not large enough for the largest report cache, that cache will not be used and the report will always execute against the warehouse. For example, if the largest report you want to be cached in memory is 20 MB, the maximum RAM usage needs to be at least 20 MB.

- The average size and number of cache files.
- The amount of memory on the Intelligence Server machine.
- The amount of memory used while the system is at maximum capacity.

You should monitor the system's performance when you change the **Maximum RAM usage** setting. In general, it should not be more than 30% of the machine's total memory.

For more information about when report caches are moved in and out of memory, see *Location of result caches, page 404*.

Maximum number of caches

The **Maximum number of caches** settings, in the Project Configuration Editor under the **Caching: Result Caches: Storage** category, limit the number of result caches, including Matching caches, History caches, Matching-History caches, and XML caches, allowed in the project at one time. The default values are 10,000 datasets, and 100,000 formatted documents.

This setting depends on the following factors:

- The number of users and the number of History List messages they keep.
- The number of report caches and their average size.
- The amount of hard disk space available in the cache directory.

RAM swap multiplier

If the Intelligence Server memory that has been allocated for caches becomes full, it must swap caches from memory to disk. The **RAM swap multiplier** setting, in the Project Configuration Editor under the **Caching: Result Caches: Storage** category, controls how much memory is swapped to disk, relative to the size of the cache being swapped into memory. For example, if the **RAM swap multiplier** setting is 2 and the requested cache is 80 kilobytes, 160 kilobytes are swapped from memory to disk.

 © 2016, MicroStrategy Inc.

If the cache memory is full and several concurrent reports are trying to swap from disk, the swap attempts can fail and re-execute those reports. This counteracts any gain in efficiency due to caching. In this case, increasing the **RAM swap multiplier** setting provides additional free memory into which those caches can be swapped.

The default value for this setting is 2.

Maximum RAM for cache index %

This setting determines what percentage of the amount of memory specified in the **Maximum RAM usage** limits (see *Maximum RAM usage, page 421*) can be used for result cache lookup tables. If your reports and documents contain many prompt answers, the cache lookup table may reach this limit. At this point, Intelligence Server no longer creates new caches. To continue creating new caches, you must either remove existing caches to free up memory for the cache lookup table, or increase this limit.

The default value for this parameter is 100%, and the values can range from 10% to 100%.

You can change this setting in the Project Configuration Editor under the **Caching: Result Caches: Storage** category.

Load caches on startup

If report caching is enabled and the **Load caches on startup** setting is enabled, when Intelligence Server starts up, it loads report caches from disk into memory until it reaches the **Maximum RAM usage** limit (see *Maximum RAM usage, page 421*). If the **Load caches on startup** setting is disabled, it loads report caches only when requested by users.

Load caches on startup is enabled by default. To disable it, in the Project Configuration Editor under the **Caching: Result Caches: Storage** category, clear the **Load caches on startup** check box.

For large projects, loading caches on startup can take a long time so you have the option to set the loading of caches on demand only. However, if caches are not loaded in advance, there will be a small additional delay in response time when they are hit. Therefore, you need to decide which is best for your set of user and system requirements.

Never expire caches

The **Never expire caches** setting, in the Project Configuration Editor under the **Caching: Result Caches: Maintenance** category, causes caches to never automatically expire. MicroStrategy recommends selecting this check box, instead of using time-based result cache expiration. For more information, see *Expiring result caches, page 416*.

Cache duration (Hours)

All caches that have existed for longer than the **Cache duration (Hours)** are automatically expired. This duration is set to 24 hours by default. You can change the duration in the

© 2016, MicroStrategy Inc.

Project Configuration Editor under the **Caching: Result Caches: Maintenance** category.

As mentioned earlier, MicroStrategy recommends against using time-based result cache expiration. For more information, see *Expiring result caches, page 416*.

Cache expiration and dynamic dates

By default, caches for reports based on filters that use dynamic dates always expire at midnight of the last day in the dynamic date filter. This behavior occurs even if the **Cache Duration** (see above) is set to zero.

For example, a report has a filter based on the dynamic date "Today." If this report is executed on Monday, the cache for this report expires at midnight on Monday. This is because a user who executes the report on Tuesday expects to see data from Tuesday, not the cached data from Monday. For more information on dynamic date filters, see the *Filters* chapter in the *MicroStrategy Advanced Reporting Guide*.

To change this behavior, in the Project Configuration Editor under the **Caching: Result Caches: Maintenance** category, select the **Do Not Apply Automatic Expiration Logic for reports containing dynamic dates** check box. When this setting is enabled, report caches with dynamic dates expire in the same way as other report caches do, according to the **Cache duration** setting.

Cache usage defaults for subscriptions

By default, if a cache is present for a subscribed report or document, the report or document uses the cache instead of re-executing the report or document. If no cache is present, one is created when the report or document is executed. For more information about subscriptions, see *Scheduling reports and documents: Subscriptions, page 488*.

When you create a subscription, you can force the report or document to re-execute against the warehouse even if a cache is present. You can also prevent the subscription from creating a new cache.

To change the default behavior for new subscriptions, use the following check boxes in the Project Configuration Editor, in the Caching: Subscription Execution category.

- To cause new History List and Mobile subscriptions to execute against the warehouse by default, select the **Re-run History List and Mobile subscriptions against the warehouse** check box.
- To cause new email, file, and print subscriptions to execute against the warehouse by default, select the **Re-run file, email, and print subscriptions against the warehouse** check box.
- To prevent new subscriptions of all types from creating or updating caches by default, select the **Do not create or update matching caches** check box.

Result cache settings at the report level

These setting allows you to disable or enable caching for a specific report or document.

 © 2016, MicroStrategy Inc.

You must have the Use Design Mode privilege to configure report/document-level cache settings.

Result caching options

- To set the caching options from a report, in the Report Editor, from the **Data** menu, select **Report caching options**. The Report Caching Options dialog box opens.
- To set the caching options for a document, in the Document Editor, from the **Format** menu, select **Document Properties**. The Document Properties dialog box opens. Select the **Caching** category.

To enable caching for a report or document even if caching is disabled at the project level, select the **Enable Caching** option.

For a document, you can choose which formats, such as HTML or PDF, are cached. You can also choose to create a new cache for every page-by, incremental fetch block, and selector setting.

To disable caching for a report or document even if caching is enabled at the project level, select the **Disable Caching** option.

To use the project-level setting for caching, select the **Use default project-level behavior** option. This indicates that the caching settings configured at the project level in the Project Configuration Editor apply to this specific report or document as well.

Saving report results: History List

The *History List* is a folder where Intelligence Server places report and document results for future reference. Each user has a unique History List.

With the History List, users can:

- Keep shortcuts to previously run reports, like the Favorites list when browsing the Internet.
- Perform asynchronous report execution. For example, multiple reports can be run at the same time within one browser, or pending reports can remain displayed even after logging out of a project.
- View the results of scheduled reports.

The History List is displayed at the user level, but is maintained at the project source level. The History List folder contains messages for all the projects in which the user is working. The number of messages in this folder is controlled by the setting **Maximum number of messages per user**. For example, if you set this number at 40, and you have 10 messages for Project A and 15 for Project B, you can have no more than 15 for Project

C. When the maximum number is reached, the oldest message in the current project is purged automatically to leave room for the new one.

If the current project has no messages but the message limit has been reached in other projects in the project source, the user may be unable to run any reports in the current project. In this case the user must log in to one of the other projects and delete messages from the History list in that project.

This section provides the following information about History Lists:

- *Understanding History Lists, page 426*
- *Configuring History List data storage, page 429*
- *Accessing History Lists, page 432*
- *Archiving History List messages, page 434*
- *Managing History Lists, page 435*

Understanding History Lists

A History List is a collection of pre-executed reports and documents that have been sent to a user's personal History folder. These pre-executed reports and documents are called History List messages.

The data contained in these History List messages is stored in the History List repository, which can be located on Intelligence Server, or in the database. For more information about the differences between these storage options, see *Configuring History List data storage, page 429*.

A History List message provides a snapshot of data at the time the message is created. Using a different report filter on a History List message does not cause the message to return different data. To view a report in the History List with a different report filter, you must re-execute the report.

Each report that is sent to the History List creates a single History List message. Each document creates a History List message for that document, plus a message for each dataset report in the document.

You can send report results to the History List manually or automatically.

Sending a message to the History List manually

Report results can be manually sent to the History List any time you plan to execute a report, during report execution, or even after a report is executed:

- **Before report execution**:
  - **From Developer**: Right-click the report or document name and select **Send to History** from the shortcut menu. The report or document is executed, and a message is generated in the History List.

 © 2016, MicroStrategy Inc.

This option is not available from a shortcut to a report or document.

- **From Web**: This option is not available.

- **In the middle of report execution**:
  - **From Developer**: While the report is being executed, select **Send to History List** from the File menu.

This operation creates two jobs, one for executing the report (against the data warehouse) and another for sending the report to History List. If caching is enabled, the second job remains in the waiting list for the first job to finish; if caching is not enabled, the second job runs against the data warehouse again. Therefore, to avoid wasting resources, MicroStrategy recommends that if caching is not enabled, users not send the report to History List in the middle of a report execution.

  - **From Web**: While the report is being executed, click **Add to History List** on the wait page.

This operation creates only one job because the first one is modified for the Send to History List request.

- **After report execution**:
  - **From Developer**: After the report is executed, select **Send to History** from the **File** menu.
  - **From Web**: After the report is executed, select **Add to History List** from the **Home** menu.

Two jobs are created for Developer, and only one is created for Web.

Sending a message to the History List automatically

Report results can be automatically sent to the History List. There are two different ways to automatically send messages to the History list. You can either have every report or document that you execute sent to your History List, or you can subscribe to specific reports or documents:

- **To automatically send every report and document that is executed to your History List:**
  - **From MicroStrategy Web**: Select **History List** from the Project Preferences, and then select **Automatically** for **Add reports and documents to my History List**.
  - **From Developer**: Select **MicroStrategy Developer Preferences** from the **Tools** menu, then select **History Options**, and then select **Automatically send reports to History List during execution**.
- **To schedule delivery of specific reports or documents:**

- **From MicroStrategy Web**: On the reports page, under the name of the report that you want to send to History List, select **Subscriptions**, and then click **Add History List subscription** on the My Subscriptions page. Choose a schedule for the report execution. A History List message is generated automatically whenever the report is executed based on the schedule.
- **From Developer**: Right-click a report or document and select **Schedule Delivery to** and select **History List**. The History List Subscription Editor opens. Define the subscription details. For specific information about using the Subscription Editor, click **Help.**

Filtering and purging your History List messages in Developer

The History List Monitor filter can be used to either filter which messages are displayed in the History List, or it can define the History List messages that you want to purge from the History List. The History List Monitor filter allows you to define various parameters to filter or purge your History List messages.

To use the History List Monitor Filter to filter your History List messages, right click the History List folder, and select **Filter**. After you have specified the filter parameters, click **OK.** The History List Monitor Filter closes, and your History List messages will be filtered accordingly.

To use the History List Monitor Filter to purge items from your History List folder, right click the History List folder and select **Purge**. The History List Monitor Filter opens. After you have specified the filter parameters, click **Purge**. The History List Monitor Filter closes, and the History List Messages that match the criteria defined in the History List Monitor Filter are deleted.

For more details about the History List Monitor Filter, click **Help**.

History Lists and caching

The History List is closely related to caching functionality. History Lists consist of messages that point to report results, which are stored as History caches. Therefore, when a History List message is deleted, the History cache that the message points to is deleted as well.

Multiple messages can point to the same History cache. In this case, the History cache is deleted after all messages pointing to it have been deleted.

If you are using a database-based History List repository, by default, duplicates of the report caches that are associated with the History List messages are stored in the database, as well as being stored locally. This way, if a user deletes the local report cache, the cache that is stored in the database can still be accessed. This behavior applies to both History Caches and History-Matching Caches. For more information about types of caches, see *Types of result caches, page 403*. For more information about storing History List data, see *Configuring History List data storage, page 429*.

If you are exporting a report or document to a database-based History List, only the most recent export is stored in the History List. For example, if you export a

 © 2016, MicroStrategy Inc.

document as an Excel file, and then export it as a PDF, only the PDF is saved in the History List.

You can use the History List messages to retrieve report results, even when report caching is disabled.

Configuring History List data storage

The History List repository is the location where all History List data is stored.

There are several different ways that the History List repository can be configured to store data for the History List. It can be stored in a database, or in a file on the Intelligence Server machine. Alternately, you can use a hybrid approach that stores the message information in a database for improved search results and scalability, and the message results in a file for performance reasons.

Configuring Intelligence Server to use a database-based or hybrid History List repository

The caches associated with History Lists can be stored in a database. Storing the History List messages in a database reduces the load on the machine that hosts Intelligence Server.

If you are using a database-based History List repository, the caches that are associated with a History List message are also stored in the History List database.

You can also configure Intelligence Server to use a hybrid History List repository. In this configuration the History List message information is stored in a database, and the cached results are stored in a file. This approach preserves the scalability of the database-based History List, while maintaining the improved performance of the file-based History List.

Be aware of the following:

- Once Intelligence Server has been configured to store the History List cached data in the database, this setting will apply to the entire server definition.
- MicroStrategy does not recommend reverting back to a file-based History List repository. If you want to revert back to a file-based repository, you must replace the existing server definition with a new one.

Prerequisites

- The storage location for the History List data (the History List repository) must have been created in the database. For information about creating the History List repository in the database, see the *MicroStrategy Installation and Configuration Guide*.
- If you are using a hybrid History List repository, the storage location for the History List results must have been created and shared on the Intelligence Server machine. For information about how to configure this location, see *Configuring Intelligence Server to use a file-based History List repository, page 431*.

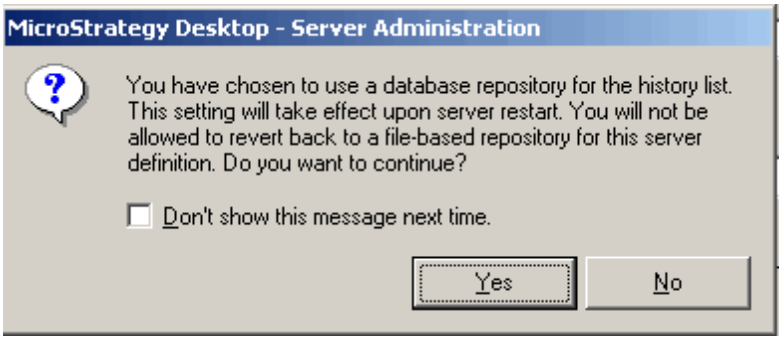
To configure Intelligence Server to use a database-based or hybrid History List repository

1 In Developer, log in to the project source as a user with administrative privileges.

2 From the **Administration** menu, select **Server,** then **Configure MicroStrategy Intelligence Server.** The Intelligence Server Configuration Editor opens.

3 On the left, expand **History Settings** and select **General.**

4 Select **Database based**. The following warning message is displayed:



Once Intelligence Server has been configured to store the History List cached data in the database, this setting will apply to the entire server definition.

5 Click **Yes.** The warning message closes.

6 By default, History List caches are backed up to the database. To store only History List caches on the server, clear the **Backup report history caches to the database** check box.

7 To use a hybrid History List repository, in the **External central storage directory for Database-based History List** field, type the location for the file-based History List message storage. For information about how the cached results are stored, see *Configuring Intelligence Server to use a file-based History List repository, page 431*.

You can browse to the file location by clicking the **. . .** (browse) button.

8 Expand **Server Definition**, and then select **General**.

9 Under **Content Server Location**, from the **Database Instance** menu, select the database instance that points to the History List repository in the database.

10 Click **OK.** The Intelligence Server Configuration Editor closes.

11 Restart Intelligence Server for the changes to take effect.

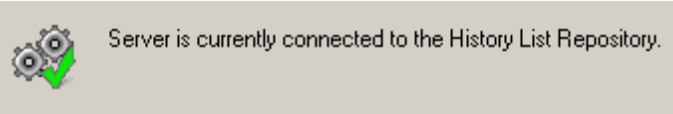
To confirm that the History List repository has been configured correctly

12 Log in to the project source as a user with administrative privileges.

 © 2016, MicroStrategy Inc.

13 From the **Administration** menu, select **Server,** then **Configure MicroStrategy Intelligence Server.** The Intelligence Server Configuration Editor opens.

14 On the left, expand **History Settings** and select **General.** If you have configured Intelligence Server properly, the following message is displayed in the Repository Type area of the Intelligence Server Configuration Editor:



Configuring Intelligence Server to use a file-based History List repository

When you initially set up your History List, you can store the History List in a file folder on the machine that hosts Intelligence Server. The default location of this folder is relative to the installation path of Intelligence Server:

```
.\Inbox\<Server definition name>
```

For example, `C:\Program Files (x86)\MicroStrategy\Intelligence Server\Inbox\MicroStrategy Tutorial Server`.

In a non-clustered environment, History List cached data is typically stored on the same machine that is running Intelligence Server.

In a clustered environment, there are two storage options:

- **Local caching:** Each node hosts its own cache file directory that needs to be shared as "ClusterCache" so that other nodes can access it.
- **Centralized caching**: All nodes have the cache file directory set to the same network location, `\\<machine name>\<shared directory name>`. For example, `\\My_File_Server\My_Inbox_Directory`.

Note the following:

For caches stored on Windows machines, and on UNIX/Linux machines using Samba, set the path to `\\<machine name>\<shared directory name>`. For caches stored on UNIX/Linux machines, set the path to `//<SharedLocation>/<CacheFolder>`.

On UNIX systems, it is recommended that you mount the shared location as a network drive. You must create a folder in your machine's `Volumes` directory before mounting the location. For example, `mount -t afp afp://my_file_server/my_inbox_directory/Volumes/my_network_mount`.

Make sure that the network directory is writable from the network account under which Intelligence Server is running. Each Intelligence Server creates its own subdirectory.

For steps to configure Intelligence Server to store cached History List data in a file-based repository, see the procedure below.

To configure Intelligence Server to use a file-based History List repository

1. Log in to the project source as a user with administrative privileges.
2. From the **Administration** menu, select **Server,** then **Configure MicroStrategy Intelligence Server.** The Intelligence Server Configuration Editor opens.
3. On the left, expand **History Settings** and select **General.**
4. Select **File based**, and type the file location in the **History directory** field.

   You can browse to the file location by clicking the **. . .** (browse) button.
5. Click **OK.** The Intelligence Server Configuration Editor closes.

Accessing History Lists

History Lists can be accessed from both MicroStrategy Web and Developer. You cannot see the History Lists for all users unless you have access to the History List Messages Monitor. For more information about the History List Messages Monitor, see *Managing History Lists, page 435*.

Accessing the History List folder in MicroStrategy Web

In MicroStrategy Web, log in to the desired project and click the **History List** link in the top navigation bar. This displays all history list messages for the user that is currently logged in. The following information is available:

- **Name**: Name (or alias) of the report.
- **Status**: Status of a report job, for example, executing, processing on another node, ready, and so on.

  If you are working in a clustered environment, only Ready and Error statuses are synchronized across nodes. While a job on one node is reported as Executing, it is reported as Processing On Another Node on all the other nodes.
- **Message Creation Time**: The time the message was created, in the currently selected time zone.
- **Details**: More information about the report, including total number of rows, total number of columns, server name, report path, message ID, report ID, status, message created, message last updated, start time, finish time, owner, report

 © 2016, MicroStrategy Inc.

description, template, report filter, view filter, template details, prompt details, and SQL statements.

Each time a user submits a report that contains a prompt, the dialog requires that he answer the prompt. As a result, multiple listings of the same report may occur. The differences among these reports can be found by checking the timestamp and the data contents.

You can export a message, view a message in PDF format, or rename a message. For more detailed instructions, see the MicroStrategy Web Help.

Accessing the History List folder in Developer

In Developer, History List messages are located in the History folder under the project name. The number next to the History List folder indicates how many unread History List messages are contained in the folder. Click the History folder to view all the messages. Each message is listed with the following information:

- **Name**: Name of the report
- **Finish Time**: The time the report execution is finished
- **Folder name**: Name of the folder where the original report is saved
- **Last update time**: The time when the original report was last updated
- **Message text**: The status message for the History List message
- **Start time**: The time the report execution was started
- **Status**: Status of a report job, for example, has been executed successfully and is ready, is not executed successfully, is currently executing, or is waiting to execute

You can see more details of any message by right-clicking it and selecting **Quick View**. This opens a new window with the following information:

- **Report definition**: Expand this category to see information about the report definition, including the description, owner, time and date it was last modified, the project it resides in, the report ID, the path to the report's location, and report details.
- **Job execution statistics**: Expand this category to see information about the report execution, including the start and end time, the total number of rows and columns in the report, the total number of rows and columns that contain raw data, whether a cache was used, the job ID, and the SQL produced.
- **Message status**: Expand this category to see information about the message itself, including the language, user creation time, last update time, read status, format, request type, application, message ID, and message text.

© 2016, MicroStrategy Inc.

Archiving History List messages

Generally, you archive History List messages if you want to see the report results as they were when the messages were originally created. This feature is useful when you need to track changes in the report results for a scheduled report.

Intelligence Server automatically marks History List messages as archived when, in the Subscription Editor, the **The new scheduled report will overwrite older versions of itself** check box is cleared. Archived messages can also be created in a MicroStrategy Web subscription if, on the Project Defaults - History List Preferences page, the **The new scheduled report will overwrite older versions of itself** check box is cleared.

To archive all History List messages in a project in Web

1. In Preferences Levels category, select **Project defaults**.
2. In the Preferences category, select **History List**.
3. Clear the check box for **The new scheduled report will overwrite older versions of itself**.

To archive History List messages in Developer

1. From the **Administration** menu, select **Scheduling** and then select **Subscription Creation Wizard**. The Subscription Creation Wizard opens.
2. Review the steps displayed, and click **Next**.
3. Specify the following characteristics of the schedule:
   - Choose the schedule that you want to use.
   - Choose the project that contains the object that you want to archive.
   - Choose **History List** from the **Delivery Type** drop-down menu.

   Click **Next**.
4. Choose the reports or documents that you wish to archive:
   - Browse to the report or document that you want to archive. You can select multiple reports or documents by holding the **Ctrl** key while clicking them.
   - Click the right arrow to add the report or document.
   - Click **Next** when all of the reports or documents that you want to archive have been added.
5. Select a user group to receive the message for the archived report or document:

 © 2016, MicroStrategy Inc.

- Browse to the user group that you want to send the archived report to. You can select multiple reports or documents by holding the **Ctrl** key while clicking them.
- Click the right arrow to add the group.
- Click **Next** when all of the user groups that you want to receive the archived report or document have been added.

All members in the user group receive the History List message.

6 Specify the subscription properties. You can choose to do the following:

- Run the schedule immediately
- Set the expiration date for the subscription
- Send a delivery notification to all users included in the subscription.

7 Clear the **The new scheduled report will overwrite older versions of itself** check box, and click **Next.**

8 Review the summary screen and click **Finish.** The Subscription Creation Wizard closes.

Managing History Lists

Administrators manage History Lists and the History caches at the same time. For information on the relationship between the History caches and History Lists, see *History caches, page 404*.

An administrator can control the size of the History List and thus control resource usage through the following settings:

- The maximum size of the History List is governed at the project level. Each user can have a maximum number of History List messages, set by an administrator. For more details, including instructions, see *Controlling the maximum size of the History List, page 436*.
- Message lifetime is the length of time before a History List message is automatically deleted. For more details about message lifetime, see *Controlling the lifetime of History List messages, page 437*.
- You can also delete History List messages according to a schedule. For more details, including instructions, see *Scheduling History List message deletion, page 437*.
- If you are using a database-based History List, you can reduce the size of the database by disabling the History List backup caches. For more details, including instructions, see *Backing up History caches to the History List database, page 438*.

If you are using a database-based History List repository and you have the proper permissions, you have access to the History List Messages Monitor. This powerful tool allows you to view and manage History List messages for all users. For more information, see *Monitoring History List messages, page 439*.

© 2016, MicroStrategy Inc.

History List backup frequency

The backup frequency for History List messages is the same as for caching. History List messages are backed up to disk as frequently as the server backup frequency setting specifies. For more information, see *Backup Frequency (minutes), page 417*.

History Lists in a clustered environment

In a clustered environment, each server maintains its own History List file. However, the same messages are retrieved and presented to the user regardless of the machine from which the History List is accessed. For complete details on History Lists in a clustered environment, see *Synchronizing History Lists, page 377*.

Controlling the maximum size of the History List

The maximum size of the History List is governed at the project level. The project administrator can set a maximum number of History List messages for each user. The default value is -1, which means that there is no maximum number of History List messages.

The administrator can also specify whether to create separate messages for each dataset report that is included in a Report Services document or to create only a message for the document itself, and whether to create messages for documents that have been exported in other formats, such as Excel or PDF. Not creating these additional History List messages can improve History List performance, at the cost of excluding some data from the History List. By default, all reports and documents create History List messages.

To configure the messages that are stored in the History List

1. In Developer, log into a project. You must log in with a user account that has administrative privileges.
2. From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.
3. Expand the **Project Definition** category and select the **History list** subcategory.
4. In the **Maximum number of messages per user** field, type the maximum number of History List messages to store for each user, or type -1 for no limit.
5. To create a History List message for each dataset report included in a Report Services document, select the **Save Report Services document dataset messages to History List** check box. To create only a message for the document, and not for the dataset reports, clear this check box.
6. To create History List messages for Report Services documents that are exported to other formats, select the **Save exported results for interactive executions sent to History List** check box. To not create messages for documents when they are exported, clear this check box.
7. Click **OK** to close the Project Configuration Editor.

 © 2016, MicroStrategy Inc.

8 Restart Intelligence Server for your changes to take effect.

Controlling the lifetime of History List messages

Message lifetime controls how long (in days) messages can exist in a user's History List. This setting allows administrators to ensure that no History List messages reside in the system indefinitely. Messages are tested against this setting at user logout and deleted if found to be older than the established lifetime.

When a message is deleted for this reason, any associated History caches are also deleted. For more information about History caches, see *History caches, page 404*.

The default value is -1, which means that messages can stay in the system indefinitely until the user manually deletes them.

To set message lifetime

1 In Developer, log into a project source.

2 From the **Administration** menu, point to **Server** and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.

3 Expand **History Settings** on the left, then select **General**.

4 Type a number in the **Message lifetime (days)** field.

5 Click **OK.** The Intelligence Server Configuration Editor closes.

Scheduling History List message deletion

You can delete History List messages using the Schedule Administration Tasks feature, which is accessed by selecting **Scheduling** from the **Administration** menu. This allows you to periodically and selectively purge History List messages of certain users and groups. You can choose to target only certain messages, including:

- Messages for a certain project or for all projects
- Messages in the History Lists of all users in a certain group
- Messages that are read or unread
- Messages that were created more than x number of days ago

The Delete History List messages feature can also be used for one-time maintenance by using a non-recurring schedule.

To schedule History List message deletion

1 In Developer, log into a project source.

2. From the **Administration** menu, select **Scheduling,** then select **Schedule Administration Tasks**. The Schedule Administration Tasks window opens.
3. Select a project from the Available Projects list.
4. Select **Delete History List messages** as the action.
5. Select a schedule from the preconfigured options, for example, at close of business (weekday), first of month, on database load, and so on.
6. Type a number in the **Lifetime (days)** box.
7. Select an option for the messages status:
   - Read
   - Unread
   - All
8. Click ... (the browse button) to select a **user/group** for which the History List messages will be deleted.
9. Click **OK**.

Cleaning up the History List Database

You can clean up the History List database using the Schedule Administration Tasks feature, which is accessed by selecting **Scheduling** from the **Administration** menu. This allows you to periodically remove orphaned entries from the database, and it allows you to remove history list messages for deleted users.

The Clean History List database feature can also be used for one-time maintenance by using a non-recurring schedule.

To schedule History List database cleanup

1. In Developer, log into a project source.
2. From the **Administration** menu, select **Scheduling,** then select **Schedule Administration Tasks**. The Schedule Administration Tasks window opens.
3. Select a project from the Available Projects list.
4. Select **Clean History List database** as the action.
5. Click **OK**.

Backing up History caches to the History List database

By default, in a database-based History List, the History caches are backed up to the database. This provides increased scalability in large systems, and increases availability to the History caches in the event of a node failure. It also allows you to set longer message

 © 2016, MicroStrategy Inc.

lifetimes for History List messages, because older History caches can be deleted from the Intelligence Server machine's hard disk and can be served by the database instead.

If you are concerned about the size of the database used for a database-based History List, you can disable the use of the database as a long-term backup for History caches.

To disable the database backup for History caches

1. In Developer, log into a project source.
2. From the **Administration** menu, point to **Server** and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **History Settings** category, and select **General**.
4. Clear the **Backup report history caches to database** check box.
5. Click **OK** to exit the Intelligence Server Configuration Editor.

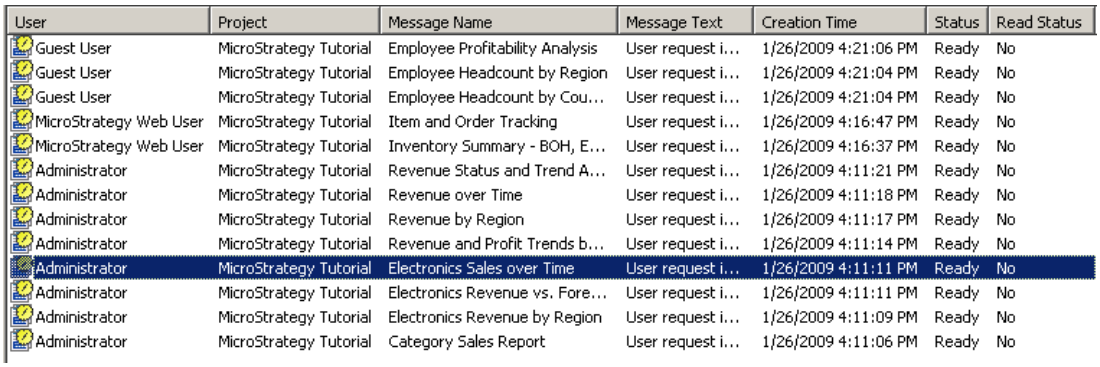
Monitoring History List messages

The History List Messages Monitor allows you to view all History List messages for all users, view detailed information about each message, and purge the messages based on certain conditions.

To use the History List Messages Monitor, your History List repository must be stored in a database. For more information about configuring the History List repository, see *Configuring Intelligence Server to use a database-based or hybrid History List repository, page 429*.

To monitor the History List messages

1. In Developer, log in to a project source. You must log in as a user with the Administer History List Monitor and the Monitor History List privileges.
2. Expand **Administration**, then expand **System Monitors**, and then select **History List Messages**. All History List messages are displayed, as shown below:



User	Project	Message Name	Message Text	Creation Time	Status	Read Status
Guest User	MicroStrategy Tutorial	Employee Profitability Analysis	User request i...	1/26/2009 4:21:06 PM	Ready	No
Guest User	MicroStrategy Tutorial	Employee Headcount by Region	User request i...	1/26/2009 4:21:04 PM	Ready	No
Guest User	MicroStrategy Tutorial	Employee Headcount by Cou...	User request i...	1/26/2009 4:21:04 PM	Ready	No
MicroStrategy Web User	MicroStrategy Tutorial	Item and Order Tracking	User request i...	1/26/2009 4:16:47 PM	Ready	No
MicroStrategy Web User	MicroStrategy Tutorial	Inventory Summary - BOH, E...	User request i...	1/26/2009 4:16:37 PM	Ready	No
Administrator	MicroStrategy Tutorial	Revenue Status and Trend A...	User request i...	1/26/2009 4:11:21 PM	Ready	No
Administrator	MicroStrategy Tutorial	Revenue over Time	User request i...	1/26/2009 4:11:18 PM	Ready	No
Administrator	MicroStrategy Tutorial	Revenue by Region	User request i...	1/26/2009 4:11:17 PM	Ready	No
Administrator	MicroStrategy Tutorial	Revenue and Profit Trends b...	User request i...	1/26/2009 4:11:14 PM	Ready	No
Administrator	MicroStrategy Tutorial	Electronics Sales over Time	User request i...	1/26/2009 4:11:11 PM	Ready	No
Administrator	MicroStrategy Tutorial	Electronics Revenue vs. Fore...	User request i...	1/26/2009 4:11:11 PM	Ready	No
Administrator	MicroStrategy Tutorial	Electronics Revenue by Region	User request i...	1/26/2009 4:11:09 PM	Ready	No
Administrator	MicroStrategy Tutorial	Category Sales Report	User request i...	1/26/2009 4:11:06 PM	Ready	No

3 To view the details of a History List message, double-click that message. A Quick View window opens, with detailed information about the message.

4 To filter the messages displayed based on criteria that you define, right-click a message and select **Filter**.

To purge a History List message

1 Select the message in the History List Monitor.

2 Right-click the message and select **Purge**. The message is purged from the database.

Element caches

When a user runs a prompted report containing an attribute element prompt or a hierarchy prompt, an element request is created. (Additional ways to create an element request are listed below.) An element request is actually a SQL statement that is submitted to the data warehouse. Once the element request is completed, the prompt can be resolved and sent back to the user. Element caching, set by default, allows for this element to be stored in memory so it can be retrieved rapidly for subsequent element requests without triggering new SQL statements against the data warehouse.

For example, if ten users run a report with a prompt to select a region from a list, when the first user runs the report, a SQL statement executes and retrieves the region elements from the data warehouse to store in an element cache. The next nine users see the list of elements return much faster than the first user because the results are retrieved from the element cache in memory. If element caching is not enabled, when the next nine users run the report, nine additional SQL statements will be submitted to the data warehouse, which puts unnecessary load on the data warehouse.

Element caches are the most-recently used lookup table elements that are stored in memory on the Intelligence Server or Developer machines so they can be retrieved more quickly. They are created when users:

 © 2016, MicroStrategy Inc.

- Browse attribute elements in Developer using the Data Explorer, either in the Folder List or the Report Editor
- Browse attribute elements in the Filter Editor
- Execute a report containing a prompt exposing an attribute list (which includes hierarchies and element list types). The element list is displayed when the report executes and creates an element cache.

This section discusses the following topics concerning element caching:

- *Element caching terminology, page 441*
- *Location of element caches, page 442*
- *Cache matching algorithm, page 442*
- *Enabling or disabling element caching, page 442*
- *Limiting the number of elements displayed and cached at a time, page 443*
- *Caching algorithm, page 446*
- *Limiting the amount of memory available for element caches, page 446*
- *Limiting which attribute elements a user can see, page 448*
- *Limiting element caches by database connection, page 449*
- *Result cache files, page 405*
- *Deleting all element caches, page 450*
- *Summary table of element cache settings, page 451*

Element caching terminology

The following terminology is helpful in understanding the concept of element caching:

- **Element Request/Browse Query**: A SQL request issued to the data warehouse to retrieve a list of attribute elements. This request accesses the attributes lookup table, which is defined when the attribute is created in Architect. If the key to the lookup table is the attribute itself, a `SELECT` is issued for the element request. If the attributes lookup table is in a lower-level lookup table (for example, month in the lookup date table) a `SELECT DISTINCT` is used for the element request. Element requests may also contain a `WHERE` clause if resulting from a search, filtered hierarchy prompt, drill request on a hierarchy prompt, or a security filter.
- **Element Cache Pool**: The amount of memory Intelligence Server allocates for element caching. In the interface, this value is called Maximum RAM usage, set in the Project Configuration Editor in the Caching: Auxiliary Caches: Elements category. The default value for this setting is 1 MB. Intelligence Server estimates that each object uses 512 bytes; therefore, by default, Intelligence Server caches about 2,048 element objects. If an element request results in more objects needing to be cached than what the maximum size of the element cache pool allows, the request is not cached.

© 2016, MicroStrategy Inc.

- **Element Incremental Fetch Size**: The maximum number of elements for display in the interface per element request. On Developer, the default for the Element Incremental Fetch setting is 1,000 elements; on Web, the default is 15 elements.

Location of element caches

Element caches are stored only in memory and are not saved to disk. They can exist on both Intelligence Server and Developer machines.

When a Developer user triggers an element request, the cache within the Developer machine's memory is checked first. If it is not there, the Intelligence Server memory is checked. If it is not there, the results are retrieved from the data warehouse. Each option is successively slower than the previous one, for example, the response time could be 1 second for Developer, 2 seconds for Intelligence Server, and 20 seconds for the data warehouse.

Cache matching algorithm

For an element cache to be used, the cache must be valid, and it must match the job being executed. The following cache keys are used in the matching process:

- Attribute ID
- Attribute version ID
- Element ID
- Search criteria
- Database connection (if the project is configured to check for the cache key)
- Database login (if the project is configured to check for the cache key)
- Security filter (if the project and attributes are configured to use the cache key)

Enabling or disabling element caching

When the MicroStrategy system is installed for the first time, the element caching is enabled by default. You can disable it for an entire project, for a Developer client, or for a specific attribute in the project's schema. The data source cache setting `DssCacheSettingElementMaxMemoryConsumption` controls the total amount of memory used by the element server cache. Setting this value to zero completely disables the element cache.

In situations where the data warehouse is loaded more that once a day, it may be desirable to disable element caching.

 © 2016, MicroStrategy Inc.

To disable element caching for a project

In the Project Configuration Editor, in the Caching: Auxiliary Caches (Elements) category, under **Server**, set the **Maximum RAM usage (KBytes)** to 0 (zero).

To disable element caching for Developer

In the Project Source Manager, select the **Memory** tab, set the **Maximum RAM usage (KBytes)** to 0 (zero).

You might want to perform this operation if you always want to use the caches on Intelligence Server. This is because when element caches are purged, only the ones on Intelligence Server are eliminated automatically while the ones in Developer remain intact. Caches are generally purged because there are frequent changes in the data warehouse that make the caches invalid.

To disable element caching for an attribute

1. In Developer, right-click the attribute and select **Edit**. The Attribute Editor opens.
2. On the Display tab, clear the **Enable element caching** check box.

Limiting the number of elements displayed and cached at a time

Incremental element fetching reduces the amount of memory Intelligence Server uses to retrieve elements from the data warehouse and improves the efficiency of Intelligence Server's element caching. You can set the maximum number of elements to display in the interface per element request in the Project Configuration Editor, by using the **Maximum number of elements to display** setting in the Project definition: Advanced category. The default value is 1,000 for Developer and 15 for Web.

Attribute element requests can be quite large (sometimes exceeding 100,000 elements). Requests of this size take a large amount of memory and time to pull into Intelligence Server and typically force many of the smaller element caches out of the element cache pool. Caching such large element lists is often unnecessary because users rarely page through extremely large element lists; they do a search instead.

When the incremental element fetching is used, an additional pass of SQL is added to each element request. This pass of SQL determines the total number of elements that exist for a given request. This number helps users decide how to browse a given attributes element list. This additional pass of SQL generates a `SELECT COUNT DISTINCT` on the lookup table of the attribute followed by a second `SELECT` statement (using an `ORDER BY`) on the same table. From the result of the first query, Intelligence Server determines if it should cache all of the elements or only an incremental set.

The incremental retrieval limit is four times the incremental fetch size. For example, if your MicroStrategy Web product is configured to retrieve 50 elements at a time, 200 elements along with the distinct count value are placed in the element cache. The user must click the next option four times to introduce another `SELECT` pass, which retrieves another 200 records in this example. Because the `SELECT COUNT DISTINCT` value was cached, this would not be issued a second time the `SELECT` statement is issued.

To optimize the incremental element caching feature (if you have large element fetch limits or small element cache pool sizes), Intelligence Server uses only 10 percent of the element cache on any single cache request. For example, if 200 elements use 20 percent of the cache pool, Intelligence Server caches only 100 elements, which is 10 percent of the available memory for element caches.

The number of elements retrieved per element cache can be set for Developer users at the project level, MicroStrategy Web product users, a hierarchy, or an attribute. Each is discussed below.

To limit the number of elements displayed for a project (affects only Developer users)

1. In Developer, log into a project. You must log in with a user account that has administrative privileges.
2. From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Project definition**, then select **Advanced**.
4. Type the limit in the **Maximum number of elements to display** box.

To limit the number of elements displayed for MicroStrategy Web product users

1. In MicroStrategy Web, log in to a project as a user with the Web Administration privilege.
2. Click the **MicroStrategy** icon, then click **Preferences**. The Web Preferences page opens.
3. Select **Project Defaults** in the Preferences Level category.
4. Select **General** in the Preferences category.
5. Type the limit for the **Maximum number of attribute elements per block** setting in the Incremental Fetch subcategory.

 © 2016, MicroStrategy Inc.

To limit the number of elements displayed on a hierarchy

1. Open the Hierarchy editor, right-click the attribute and select **Element Display** from the shortcut menu, and then select **Limit**. The Limit dialog box opens.
2. Type a number in the **Limit** box.

To limit the number of elements displayed for an Attribute

1. Open the Attribute Editor.
2. Select the **Display** tab.
3. In the **Element Display** category, select the **Limit** option and type a number in the box.

The element display limit set for hierarchies and attributes may further limit the number of elements set in the project properties or Web preferences. For example, if you set 1,000 for the project, 500 for the attribute, and 100 for the hierarchy, Intelligence Server retrieves only 100 elements.

Optimizing element requests

You may find the incremental element fetching feature's additional `SELECT COUNT DISTINCT` query to be costly on your data warehouse. In some cases, this additional query adds minutes to the element browse time, making this performance unacceptable for production environments.

To make this more efficient, you can set a VLDB option to control how the total rows are calculated. The default is to use the `SELECT COUNT DISTINCT`. The other option is to have Intelligence Server loop through the table after the initial `SELECT` pass, eventually getting to the end of the table and determining the total number of records. You must decide whether to have the database or Intelligence Server determine the number of element records. MicroStrategy recommends that you use Intelligence Server if your data warehouse is heavily used, or if the `SELECT COUNT DISTINCT` query itself adds minutes to the element browsing time.

Using Intelligence Server to determine the total number of element records results in more traffic between Intelligence Server and the data warehouse.

Either option uses significantly less memory than what is used without incremental element fetching enabled. Using the count distinct option, Intelligence Server retrieves four times the incremental element size. Using the Intelligence Server option retrieves four times the incremental element size, plus additional resources needed to loop through the table. Compare this to returning the complete result table (which may be as large as 100,000 elements) and you will see that the memory use is much less.

The setting is called **Attribute Element Number Count Method**.

To configure Attribute Element Number Count Method

1. In the Database Instance manager, select the database instance.
2. From the **Administration** menu, select **VLDB Properties**. The VLDB Properties editor opens.
3. Under **Query Optimizations**, select **Attribute Element Number Count Method** and on the right-hand side, select one of the options:
   - To have the data warehouse calculate the count, select **Use Count(Attribute@ID) to calculate total element number (will use count distinct if necessary)** -- Default
   - To have Intelligence Server calculate the count, select **Use ODBC cursor to calculate total element number**
4. Click **Save and Close**.

Caching algorithm

The cache behaves as though it contains a collection of blocks of elements. Each cached element is counted as one object and each cached block of elements is also counted as an object. As a result, a block of four elements are counted as five objects, one object for each element and a fifth object for the block. However, if the same element occurs on several blocks it is counted only once. This is because the element cache shares elements between blocks.

The cache uses the “least recently used” algorithm on blocks of elements. That is, when the cache is full, it discards the blocks of elements that have been in the cache for the longest time without any requests for the blocks. Individual elements, which are shared between blocks, are discarded when all the blocks that contain the elements have been discarded. Finding the blocks to discard is a relatively expensive operation. Hence, the cache discards one quarter of its contents each time it reaches the maximum number of allowed objects.

Limiting the amount of memory available for element caches

You can control the amount of memory that element caches use on both Intelligence Server (set at the project level) and the Developer machines. This memory is referred to as the cache pool. If Intelligence Server attempts to cache a new element request, but there is not enough available cache pool space to store all of the new elements, existing elements must be removed from memory before the new ones can be cached. When this happens, the least recently used 25% of element caches are removed from the cache.

You can configure the memory setting for both the project and the client machine in the Cache: Element subcategory in the Project Configuration Editor. You should consider these factors before configuring it:

 © 2016, MicroStrategy Inc.

- The number of attributes that users browse elements on, for example, in element prompts, hierarchy prompts, and so on
- The number of unique elements

  For example, attribute “Year” (10 years = 10 elements), attribute “city” (500 cities = 500 elements)
- Time and cost associated with running element requests on the data warehouse

  For example, if the element request for cities runs quickly (say in 2 seconds), it may not have to exist in the element cache.
- The amount of RAM on the Intelligence Server machine

To set the RAM available for element caches for a project

1. In Developer, log into a project. You must log in with a user account that has administrative privileges.
2. From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Caching**, expand **Auxiliary Caches**, then click **Elements**.
4. Specify the amount of RAM (in megabytes) in the **Server: Maximum RAM usage (MBytes)** box.

   Note the following:
   - The default value is 1 megabyte.
   - If you set the value to 0, element caching is disabled.
   - If you set it to -1, Intelligence Server uses the default value of 1 MB.
5. Specify the amount of RAM (in megabytes) in the **Maximum RAM usage (MBytes)** box.

The new settings take affect only after Intelligence Server is restarted.

To set the RAM available for element caches on Developer

1. In the Project Source Manager, click the **Caching** tab and within the Element Cache group of controls, select the **Use custom value** option.

If you select the **Use project default** option, the amount of RAM will be the same as specified in the Client section in the Project Configuration Editor described above.

2 Specify the RAM (in megabytes) in the Client section in the **Maximum RAM usage (MBytes)** field.

Limiting which attribute elements a user can see

You can limit the attribute elements that a user can see to only the elements allowed by his or her security filter. For example, if a user's security filter allows him or her to see only the Northeast Region and she runs a report that prompts for cities, only those cities in the Northeast are displayed.

This functionality can be enabled for a project and limits the element cache sharing to only those users with the same security filter. This can also be set for attributes. That is, if you do not limit attribute elements with security filters for a project, you can enable it for certain attributes. For example, if you have Item information in the data warehouse available to external suppliers, you could limit the attributes in the Product hierarchy with a security filter. This is done by editing each attribute. This way, suppliers can see their products, but not other suppliers' products. Element caches not related to the Product hierarchy, such as Time and Geography, are still shared among users.

For more information on security filters, see *Restricting access to data: Security filters, page 79*.

To limit which attribute elements users can see per project

1 In Developer, log into a project. You must log in with a user account that has administrative privileges.

2 From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.

3 Expand **Project definition**, then select **Advanced**.

4 Select the **Apply security filters to element browsing** check box.

To limit which attribute elements users can see per attribute

1 Edit the attribute, and click the **Display** tab.

2 Select the **Apply security filters to element browsing** check box.

You must update the schema before changes to this setting take affect (from the **Schema** menu, select **Update Schema**).

 © 2016, MicroStrategy Inc.

Limiting element caches by database connection

In most cases, users connect to the data warehouse based on their connection maps. By default, all users have the same connection map, unless you map them to different ones with the Connection Mapping editor. When using connection mapping, you can also ensure that users with different database connections cannot share element caches. This causes the element cache matching key to contain the user's database connection.

To limit element caches by database connection

1. In Developer, log into a project. You must log in with a user account that has administrative privileges.
2. From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Caching**, expand **Auxiliary Caches**, then click **Elements**.
4. Select the **Create element caches per connection map** check box.

The new setting takes affect only after the project is reloaded or after Intelligence Server is restarted.

For more information about connection mapping, see *Controlling access to the database: Connection mappings, page 74*.

Users may connect to the data warehouse using their linked warehouse logins, as described below.

Limiting element caches by database login

This setting allows you to ensure that users with different data warehouse logins cannot share element caches. When this feature is used, the element cache matching key contains the user's database login. Only users with the same database login are able to share the element caches. Before you enable this feature, you must configure two items.

1. The user must have a **Warehouse Login** and **Password** specified (selecting the Authentication tab in the User Editor).
2. The project must be configured to **Use linked warehouse login for execution** (in the Project Configuration Editor, select the **Project definition: Advanced** category).

If both of these properties are not set, the users will use their connection maps to connect to the database.

© 2016, MicroStrategy Inc.

To limit element caches by database login

1. In Developer, log into a project. You must log in with a user account that has administrative privileges.
2. From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Caching**, expand **Auxiliary Caches**, then select **Elements**.
4. Select the **Create element caches per passthrough login** check box.

The new setting takes affect only after the project is reloaded or after Intelligence Server is restarted.

Deleting all element caches

You can purge (delete) all of the element caches for a project on both the Developer and Intelligence Server machines. This does not delete element caches on other Developer machines. You cannot delete only certain caches; they are all deleted at the same time.

If you are using a clustered Intelligence Server setup, to purge the element cache for a project, you must purge the cache from each node of the cluster individually.

Even after purging element caches, reports and documents may continue to display cached data. This can occur because results may be cached at the report/document and object levels in addition to at the element level. To ensure that a re-executed report or document displays the most recent data, purge all three caches. For instructions on purging result and object caches, see *Purging all result caches in a project, page 415* and *Deleting object caches, page 454*.

To delete all element caches for a project

1. In Developer, log into a project. You must log in with a user account that has administrative privileges.
2. From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Caching**, expand **Auxiliary Caches**, then select **Elements**.
4. Click **Purge Now**.

All element caches are automatically purged whenever schema is updated.

 © 2016, MicroStrategy Inc.

Summary table of element cache settings

Many of the settings that help make element caching an efficient use of system resources are explained in the sections above.

The following table lists all MicroStrategy's element caching settings.

Setting	For information...
Maximum number of elements to display	see *Limiting the number of elements displayed and cached at a time, page 443*
Attribute element number count method	see *Optimizing element requests, page 445*
Element cache - Max RAM usage (MBytes) Project	see *Limiting the amount of memory available for element caches, page 446*
Element cache - Max RAM usage (MBytes) Developer	see *Limiting the amount of memory available for element caches, page 446*
Apply security filter to element browsing	see *Limiting which attribute elements a user can see, page 448*
Create caches per connection map	see *Limiting element caches by database connection, page 449*
Create caches per passthrough login	see *Result cache files, page 405*
Purge element caches	see *Deleting all element caches, page 450*

Object caches

When you or any users browse an object definition (attribute, metric, and so on), you create what is called an object cache. An *object cache* is a recently used object definition stored in memory on Developer and Intelligence Server. You browse an object definition when you open the editor for that object. You can create object caches for applications.

For example, when a user opens the Report Editor for a report, the collection of attributes, metrics, and other user objects displayed in the Report Editor compose the report's definition. If no object cache for the report exists in memory on Developer or Intelligence Server, the object request is sent to the metadata for processing.

The report object definition retrieved from the metadata and displayed to the user in the Report Editor is deposited into an object cache in memory on Intelligence Server and also on the Developer of the user who submitted the request. As with element caching, any time the object definition can be returned from memory in either the Developer or Intelligence Server machine, it is faster than retrieving it from the metadata database.

© 2016, MicroStrategy Inc.

So when a Developer user triggers an object request, the cache within the Developer machine's memory is checked first. If it is not there, the Intelligence Server memory is checked. If the cache is not even there, the results are retrieved from the metadata database. Each option is successively slower than the previous. If a MicroStrategy Web product user triggers an object request, only the Intelligence Server cache is checked before getting the results from the metadata database.

This section discusses the following topics concerning object caching:

- *Cache matching algorithm, page 452*
- *Enabling or disabling object caching, page 452*
- *Limiting the amount of memory available for object caches, page 452*
- *Deleting object caches, page 454*
- *Summary table of object caching settings, page 455*

Cache matching algorithm

For an object cache to be used, the cache must be valid, and it must match the job being executed. The following cache keys are used in the matching process:

- Object ID
- Object version ID
- Project ID

Enabling or disabling object caching

Object caching is enabled by default when the MicroStrategy system is first installed. Object caching cannot be disabled. Intelligence Server must maintain a minimum amount of memory (1 MB) available for the object caches to operate efficiently.

Limiting the amount of memory available for object caches

You can control the amount of memory that object caches can use on both Intelligence Server (set at the project level) and the Developer machines. This memory is referred to as the cache pool. If a new object request size is small enough to fit into the object cache pool, but there is not enough available cache pool space to store all of the new objects, existing objects must be removed from memory before the new ones can be cached. When this happens, the least recently used 25% of object caches are removed from the cache.

This setting depends on the following factors:

- Size of the project in terms of application objects

 © 2016, MicroStrategy Inc.

- The amount of RAM on the Intelligence Server machine

For a project that has a large schema object, the project loading speed suffers if the maximum memory for object cache setting is not large enough. This issue is recorded in the `DSSErrors.log` file. See MicroStrategy Tech Note TN4205 for more information.

You maintain object caching by using the **Server**: **Maximum RAM usage (MBytes)** setting in the Caching: Auxiliary Caches (Objects) subcategory in the Project Configuration Editor. On the client machine, you maintain object caching by using the **Client**: **Maximum RAM usage (MBytes)** setting in the Caching: Auxiliary Caches (Objects) subcategory in the Project Configuration Editor.

The default values for **Server: Maximum RAM Usage** and **Client: Maximum RAM Usage** are 50 MB. Intelligence Server estimates that each object consumes 5 KB of the cache pool, so by default it caches 10,240 objects in server memory and 10,240 objects in client memory.

To set the RAM available for object caches for a project

1. In Developer, log into a project. You must log in with a user account that has administrative privileges.
2. From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Caching**, expand **Auxiliary Caches**, then select **Objects**.
4. Specify the RAM (in megabytes) in the Server section in the **Maximum RAM usage (MBytes)** box.
5. Specify the RAM (in megabytes) in the Client section in the **Maximum RAM usage (MBytes)** box.

The new settings take affect only after Intelligence Server is restarted.

To set the RAM available for object caches for a Developer machine

1. In the Project Source Manager, click the **Caching** tab and in the Object Cache group of controls, select the **Use custom value** option.

If you select the **Use project default** option, the amount of RAM is the same as specified in the Client section in the Project Configuration Editor described above.

2. Specify the RAM (in megabytes) in the **Maximum RAM usage (MBytes)** box.

Deleting object caches

You can purge (delete) all of the object caches for a project on both the Developer and Intelligence Server machines. However, this does not delete object caches on other Developer machines. You cannot select to delete only certain object caches; they are all deleted at the same time.

Even after purging object caches, reports and documents may continue to display cached data. This can occur because results may be cached at the report/document and element levels, in addition to at the object level. To ensure that a re-executed report or document displays the most recent data, purge all three caches. For instructions on purging result and element caches, see *Purging all result caches in a project, page 415* and *Deleting all element caches, page 450*.

To delete all object caches for a project

1. In Developer, log into a project. You must log in with a user account that has administrative privileges.
2. From the **Administration** menu, point to **Projects**, and then select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Caching**, expand **Auxiliary Caches**, then select **Objects**.
4. Click **Purge Now**.

Object caches are automatically purged whenever your schema is updated.

Configuration objects are cached at the server level. You can choose to delete these object caches as well.

To delete all configuration object caches for a server

1. Log in to the project source.
2. From the **Administration** menu in Developer, point to **Server**, and then select **Purge Server Object Caches**.

You cannot automatically schedule the purging of server object caches from within Developer. However, you can compose a Command Manager script to purge server object caches and schedule that script to execute at certain times. For a description of this process, see MicroStrategy Tech Note TN12270. For more information about Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*.

 © 2016, MicroStrategy Inc.

Summary table of object caching settings

Many of the settings that help make object caching an efficient use of system resources are explained in the sections above. The table below lists all MicroStrategy object caching settings.

Setting	For information...
Object cache - Max RAM usage (MBytes) Project level	See *Limiting the amount of memory available for object caches, page 452*
Object cache - Max RAM usage (MBytes) Developer	See *Limiting the amount of memory available for object caches, page 452*
Purge object caches	See *Deleting object caches, page 454*

 © 2016, MicroStrategy Inc.

11

MANAGING INTELLIGENT CUBES

Introduction

You can return data from your data warehouse and save it to Intelligence Server memory, rather than directly displaying the results in a report. This data can then be shared as a single in-memory copy, among many different reports created by multiple users. The reports created from the shared sets of data are executed against the in-memory copy, also known as an *Intelligent Cube*, rather than having to be executed against a data warehouse.

Intelligent Cubes are part of the OLAP Services feature in Intelligence Server. For detailed information about Intelligent Cubes, see the *MicroStrategy In-memory Analytics Guide*.

This section contains the following information:

- *Managing Intelligent Cubes: Intelligent Cube Monitor, page 458*
- *Governing Intelligent Cube memory usage, loading, and storage, page 464*
- *Supporting connection mappings in Intelligent Cubes, page 474*

© 2016, MicroStrategy Inc.

Managing Intelligent Cubes: Intelligent Cube Monitor

You must create Intelligent Cubes before they can be published. For information on creating Intelligent Cubes, see the *In-memory Analytics Guide*.

Once an Intelligent Cube has been published, you can manage it from the Intelligent Cube Monitor. You can view details about your Intelligent Cubes such as last update time, hit count, memory size, and so on.

If you are running Intelligence Server on HP-UX v2, you may notice a slow response time when using the Intelligent Cube Monitor. For information about this delay, including steps you can take to improve performance, see *Cache Monitor and Intelligent Cube Monitor performance, page 767*.

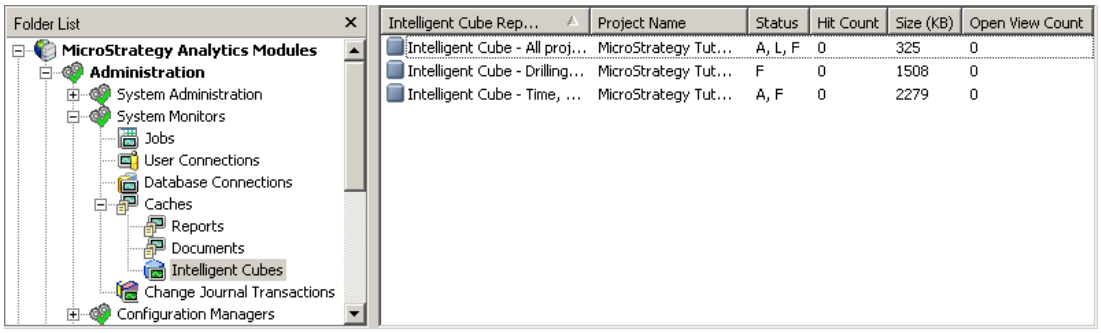
To view the available Intelligent Cubes

1. In Developer, log in to a project source. You must log in as a user with the Monitor Cubes privilege.
2. Expand **Administration**, then expand **System Monitors**, **Caches**, and then select **Intelligent Cubes**. Information about the existing Intelligent Cubes displays on the right-hand side.
3. To view the details of an Intelligent Cube, double-click that Intelligent Cube. A Quick View window opens, with detailed information about the Intelligent Cube.
4. To change the status of an Intelligent Cube, right-click that Intelligent Cube and select the desired action from the list. For a description of all Intelligent Cube statuses, see *Monitoring and modifying Intelligent Cube status, page 460*.

Viewing Intelligent Cube information and usage statistics

The Intelligent Cube Monitor provides information about published Intelligent Cubes, as well as Intelligent Cube usage statistics. The Intelligent Cube Monitor is shown in the image below.

 © 2016, MicroStrategy Inc.



You can view the following information in the Intelligent Cube Monitor:

- **Cube Report Name**: The name of the Intelligent Cube.
- **Project Name**: The project the Intelligent Cube belongs to.
- **Status**: The current status of the Intelligent Cube. For information on reviewing and modifying the status of an Intelligent Cube, see *Monitoring and modifying Intelligent Cube status, page 460*.
- **Last Update Time**: The time when the Intelligent Cube was last updated against the data warehouse.
- **Last Update Job**: The job number that most recently updated the Intelligent Cube against the data warehouse. You can use the Job Monitor to view information on a given job.
- **Creation Time**: The time when the Intelligent Cube was first published to Intelligence Server.
- **Size (KB)**: The size of the Intelligent Cube, in kilobytes.
- **Hit Count**: The number of times the Intelligent Cube has been used by reports since it was last loaded into Intelligence Server's memory. You can reset the Hit Count to zero by unloading the Intelligent Cube from Intelligence Server's memory.
- **Historic Hit Count**: The total number of times the Intelligent Cube has been used by reports. You can reset the Historic Hit Count to zero by deleting the Intelligent Cube's cache, and then republishing the Intelligent Cube.
- **Open View Count**: The number of reports currently accessing the Intelligent Cube.
- **Owner**: The user who published the Intelligent Cube.
- **Database connection**: The database connection account used for the Intelligent Cube to run against the data warehouse.
- **File Name**: The file location where the Intelligent Cube is saved to the machine's secondary storage.
- **Cube Instance ID**: The ID for the current published version of the Intelligent Cube.
- **Cube Definition ID**: The ID for the Intelligent Cube object.

- **Data Language**: The language used for the Intelligent Cube. This is helpful if the Intelligent Cube is used in an internationalized environment that supports multiple languages.
- **Total number of rows**: The number of rows of data that the Intelligent Cube contains. To view this field, the Intelligent Cube must be published at least once.
- **Total number of columns**: The number of columns of data that the Intelligent Cube contains. To view this field, the Intelligent Cube must be published at least once.

You can also view Intelligent Cube information for a specific Intelligent Cube, by double-clicking that Intelligent Cube in the Intelligent Cube Monitor. This opens a Quick View of the Intelligent Cube information and usage statistics.

Monitoring and modifying Intelligent Cube status

The status of an Intelligent Cube tells you how the Intelligent Cube is currently being used and whether reports can access the Intelligent Cube. To modify the status of an Intelligent Cube, right-click the Intelligent Cube in the Intelligent Cube Monitor, and select one of the actions listed below:

Action	Required Status to Perform Action	Description
Activate	Filed, but not Active	Loads a previously deactivated Intelligent Cube as an accessible set of data for multiple reports.
Deactivate	Active	Removes an Intelligent Cube instance from Intelligence Server memory, but saves it to secondary storage, such as a hard disk.
Update	Active	Re-executes and publishes an Intelligent Cube. When the data for an Intelligent Cube is modified and saved, the Update action updates the Intelligent Cube with the latest data.
Save to disk	Loaded	Saves an Intelligent Cube to secondary storage, and keeps the Intelligent Cube in Intelligence Server memory. If you have defined the backup frequency as zero minutes, Intelligent Cubes are automatically saved to secondary storage, as described in *Defining when Intelligent Cubes are automatically saved to secondary storage, page 473*.
Delete	Always available	Removes a published Intelligent Cube as an accessible set of data for multiple reports. **Warning**: This action does not delete the Intelligent Cube object saved in a MicroStrategy project. To delete an Intelligent Cube object, you must log into the project containing the Intelligent Cube and delete it there. For information on whether you should deactivate or unpublish an Intelligent Cube, see *Deactivating or unpublishing an Intelligent Cube, page 461*.

 © 2016, MicroStrategy Inc.

Action	Required Status to Perform Action	Description
Load in memory	Active, but not Loaded	Moves an Intelligent Cube from your machine's secondary storage to Intelligence Server memory. For information on when to load and unload Intelligent Cubes, see *Loading and unloading Intelligent Cubes, page 462*. **Note**: If the memory limit is reached, this action unloads a previously loaded Intelligent Cube from Intelligence Server memory.
Unload from memory	Loaded	Moves an Intelligent Cube from Intelligence Server memory to your machine's secondary storage, such as a hard disk. For information on when to load and unload Intelligent Cubes, see *Loading and unloading Intelligent Cubes, page 462*.

Additional statuses such as Processing and Load Pending are also used by the Intelligent Cube Monitor. These statuses denote that certain tasks are currently being completed.

Additionally, if you have defined the backup frequency as greater than zero minutes (as described in *Defining when Intelligent Cubes are automatically saved to secondary storage, page 473*), the following additional statuses can be encountered:

- **Dirty**: This status occurs if the copy of an Intelligent Cubes data in secondary storage is not up to date with data in Intelligence Server memory. This can occur if an Intelligent Cube is updated in Intelligence Server memory but the new data is not saved to secondary storage.
- **Monitoring information dirty**: This status occurs if Intelligent Cube monitoring information changes, and this information is not updated in secondary storage. Monitoring information includes details such as the number of reports that have accessed the Intelligent Cube.

In both scenarios listed above, the data and monitoring information saved in secondary storage for an Intelligent Cube is updated based on the backup frequency. You can also manually save an Intelligent Cube to secondary storage using the Save to disk action listed in the table above, or by using the steps described in *Storing Intelligent Cubes in secondary storage, page 473*.

Deactivating or unpublishing an Intelligent Cube

Both deactivating and unpublishing an Intelligent Cube prevent reports that access an Intelligent Cube from being able to load the Intelligent Cube into Intelligence Server memory. This gives you more administrative control of when to make an Intelligent Cube available to reports.

However, each of these actions provides this administrative control in slightly different ways that can fit different scenarios.

Deactivating an Intelligent Cube saves the Intelligent Cube to secondary storage, such as a hard disk. When you reactivate the Intelligent Cube, the copy in secondary storage is loaded back into Intelligence Server memory. This option is ideal when an Intelligent Cube should not be used for some length of time, but after that should be available again in its current form.

Unpublishing an Intelligent Cube deletes the copy of data from Intelligence Server memory without making a copy of the data. To make the Intelligent Cube accessible to reports, the Intelligent Cube must be re-executed against the data warehouse and published to the Intelligent Cube Monitor. This option is ideal if the current Intelligent Cube should not be reported on until it is re-executed against the data warehouse.

Loading and unloading Intelligent Cubes

When an Intelligent Cube is published, by default, it is automatically loaded into Intelligence Server memory.

Intelligent Cubes must be loaded in Intelligence Server memory to allow reports to access the data in the Intelligent Cube. If an Intelligent Cube is constantly in use, it should be loaded in Intelligence Server memory.

Using the Intelligent Cube Monitor you can load an Intelligent Cube into Intelligence Server memory, or unload it to secondary storage, such as a disk drive.

By default, Intelligent Cubes are loaded when Intelligent Cubes are published and when Intelligence Server starts. To change these behaviors, see:

- *Publishing Intelligent Cubes without loading them into Intelligence Server memory, page 463*
- *Loading Intelligent Cubes when Intelligence Server starts, page 471*

Be aware of the following:

- If loading an Intelligent Cube into Intelligence Server memory causes the memory limit to be exceeded, a different Intelligent Cube is unloaded from Intelligent Server memory.
- The act of loading an Intelligent Cube can require memory resources up to twice the size of an Intelligent Cube. This can affect performance of your Intelligence Server as well as the ability to load the Intelligent Cube. For information on how to plan for these memory requirements, see *Planning for Intelligent Cube publishing and loading , page 466*.

One way to free memory on Intelligence Server, which can improve Intelligence Server performance, is to temporarily unload an Intelligent Cube from memory. This can be a good option for Intelligent Cubes that are not constantly in use, because when a report accessing an active but unloaded Intelligent Cube is executed, that Intelligent Cube is automatically loaded into Intelligent Server memory. Be aware that if the Intelligent Cube is very large there may be some delay in displaying report results while the Intelligent Cube is being loaded into memory. For more suggestions on how to manage Intelligence Server's memory usage, see *Chapter 8, Tuning Your System for Best Performance*.

 © 2016, MicroStrategy Inc.

Publishing Intelligent Cubes without loading them into Intelligence Server memory

By default, Intelligent Cubes are automatically loaded into Intelligence Server memory so that reports can access and analyze their data.

To conserve Intelligence Server memory, you can define Intelligent Cubes to only be stored in secondary storage when the Intelligent Cube is published. The Intelligent Cube can then be loaded into Intelligence Server memory manually, using a schedule, or whenever a report attempts to access the Intelligent Cube.

The steps below show you how to define whether publishing Intelligent Cubes loads them into Intelligence Server memory. You can enable this setting at the project level, or for individual Intelligent Cubes.

To define whether publishing Intelligent Cubes loads them into Intelligence Server memory, at the project level

1. In Developer, log in to a project using an account with administrative privileges.
2. Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Intelligent Cubes**, then select General.
4. You can select or clear the **Load Intelligent Cubes into Intelligence Server memory upon publication** check box:
   - Select this check box to load Intelligent Cubes into Intelligence Server memory when the Intelligent Cube is published. Intelligent Cubes must be loaded into Intelligence Server memory to allow reports to access and analyze their data.
   - To conserve Intelligence Server memory, you can clear this check box to define Intelligent Cubes to only be stored in secondary storage upon being published. The Intelligent Cube can then be loaded into Intelligence Server memory manually, using schedules, or whenever a report attempts to access the Intelligent Cube.

   If you are using multiple Intelligence Servers in a clustered environment, this setting applies to all nodes.
5. Click **OK** to save your changes and close the Project Configuration Editor.
6. For any changes to take effect, you must restart Intelligence Server. For clustered environments, separate the restart times for each Intelligence Server by a few minutes.

© 2016, MicroStrategy Inc.

To define whether publishing Intelligent Cubes loads them into Intelligence Server memory, at the Intelligent Cube level

1. In Developer, log in to a project using an account with administrative privileges.
2. In the Folder List, browse to the folder that contains the Intelligent Cube you want to configure.
3. Right-click the Intelligent Cube, and choose **Edit**. The Intelligent Cube Editor opens.
4. From the **Data** menu, select **Configure Intelligent Cube**. The Intelligent Cube Options dialog box opens.
5. Under the **Options** category, select **General**.
6. Clear **Use default project-level settings**.
7. Select or clear the **Load Intelligent Cubes into Intelligence Server memory upon publication** check box:
   - Select this check box to load the Intelligent Cube into Intelligence Server memory when the Intelligent Cube is published. Intelligent Cubes must be loaded into Intelligence Server memory to allow reports to access and analyze their data.
   - To conserve Intelligence Server memory, clear this check box to define Intelligent Cubes to only be stored in secondary storage upon being published. The Intelligent Cube can then be loaded into Intelligence Server memory manually, using schedules, or whenever a report attempts to access the Intelligent Cube.

   If you are using multiple Intelligence Servers in a clustered environment, this setting applies to all nodes.
8. Click **OK** to close the Intelligent Cube Options dialog box.
9. In the Intelligent Cube Editor, click **Save and Close**.
10. Restart Intelligence Server. For clustered environments, separate the restart times for each Intelligence Server by a few minutes.

Governing Intelligent Cube memory usage, loading, and storage

Intelligent Cubes must be stored in Intelligence Server memory for reports to access their data. While this can improve performance of these reports, loading too much data onto Intelligence Server memory can have a negative impact on Intelligence Server's

 © 2016, MicroStrategy Inc.

ability to process jobs. For this reason, it is important to govern how much Intelligent Cube data can be stored on Intelligence Server.

Intelligent Cube data can also be stored in secondary storage, such as a hard disk, on the machine hosting Intelligence Server. These Intelligent Cubes can be loaded into memory when they are needed. For more information, see *Loading and unloading Intelligent Cubes, page 462*.

- *Governing Intelligent Cube memory usage, page 465*
- *Loading Intelligent Cubes when Intelligence Server starts, page 471*
- *Storing Intelligent Cubes in secondary storage, page 473*

Governing Intelligent Cube memory usage

Intelligent Cubes must be stored in Intelligence Server memory for reports to access their data. While this can improve performance of these reports, loading too much data onto Intelligence Server memory can negatively affect Intelligence Server's ability to process jobs. For this reason, it is important to limit how much Intelligent Cube data can be stored on Intelligence Server.

Determining memory limits for Intelligent Cubes

Storing Intelligent Cubes can greatly improve performance by allowing reports to execute against an in memory copy of data. However, storing too much Intelligent Cube data in memory can cause other Intelligence Server processes to compete for system resources and may cause degradations in performance. This makes defining a memory limit for Intelligent Cubes an important step in maintaining Intelligence Server response time.

An Intelligent Cube memory limit defines the maximum amount of RAM of the Intelligence Server machine that can be used to store loaded Intelligent Cubes. This data is allocated separately of memory used for other Intelligence Server processes.

For example, you define a memory limit on Intelligent Cubes to be 512MB. You have 300MB of Intelligent Cube data loaded into Intelligence Server memory, and normal processing of other Intelligence Server tasks uses 100MB of memory. In this scenario, Intelligence Server uses 400MB of the RAM available on the Intelligence Server machine. This scenario demonstrates that to determine a memory limit for Intelligent Cubes, you must consider the below factors:

- The amount of RAM available on the Intelligence Server machine, and of that RAM what percentage can be used by Intelligence Server without negatively affecting performance and successful operation of the host machine.
- The average and peak usage of RAM by Intelligence Server processes other than Intelligent Cube storage. For information on setting governing limits on other Intelligence Server processes and monitoring system usage, see the *Supplemental Reference for System Administration*.
- The amount of memory required to load all Intelligent Cubes necessary to meet reporting requirements. To help save space, Intelligent Cubes that are not used often

© 2016, MicroStrategy Inc.

can be unloaded until they are required by reports (see *Loading and unloading Intelligent Cubes, page 462*).

- The Maximum RAM usage (Mbytes) memory limit can be defined per project. If you have multiple projects that are hosted from the same Intelligence Server, each project may store Intelligent Cube data up to its memory limit.

  For example, you have three projects and you set their Maximum RAM usage (Mbytes) limits to 1 GB, 1 GB, and 2 GB. This means that 4 GB of Intelligent Cube data could be stored in RAM on the Intelligence Server machine if all projects reach their memory limits.

- The size of the Intelligent Cubes that are being published and loaded into memory. The process of publishing an Intelligent Cube can require memory resources in the area of two to four times greater than the Intelligent Cube's size. This can affect performance of your Intelligence Server and the ability to publish the Intelligent Cube. For information on how to plan for these memory requirements, see the next section.

- If your project and Intelligent Cubes support multiple languages, each language supported may require additional memory.

- To help reduce Intelligent Cube memory size, review the best practices described in *Best practices for reducing Intelligent Cube memory size, page 466* below.

Best practices for reducing Intelligent Cube memory size

MicroStrategy recommends the following best practices to reduce the memory size of your Intelligent Cubes:

- Attributes commonly use numeric values for their ID forms. Using attributes defined in this way can save space as compared to attributes that use character strings for their ID forms.

- Attribute forms should be included only as required because including additional attribute forms in Intelligent Cubes requires additional memory.

- You should avoid including compound metrics and smart metrics in Intelligent Cubes. The same results provided by compound metrics and smart metrics can often be provided by creating derived metrics in reports that connect to Intelligent Cubes.

- You can define Intelligent Cube normalization to reduce the amount of memory required for an Intelligent Cube. Intelligent Cube normalization can be configured using the Data population for Intelligent Cubes VLDB property. For instructions on setting VLDB properties, see the *VLDB Properties* chapter in the *Supplemental Reference for System Administration*.

Planning for Intelligent Cube publishing and loading

Publishing an Intelligent Cube can require memory resources in the area of two to four times greater than the size of an Intelligent Cube. Once the Intelligent Cube is published, the extra resources are returned to the system and only the space required for the Intelligent Cube and some indexing information is taken up in RAM. However, you

 © 2016, MicroStrategy Inc.

should consider this peak in memory usage when planning for the publication of Intelligent Cubes.

By default, publishing an Intelligent Cube includes the step of loading the Intelligent Cube into memory. You can modify this default behavior as described in *Publishing Intelligent Cubes without loading them into Intelligence Server memory, page 463*. Loading and unloading Intelligent Cubes is described in *Loading and unloading Intelligent Cubes, page 462*.

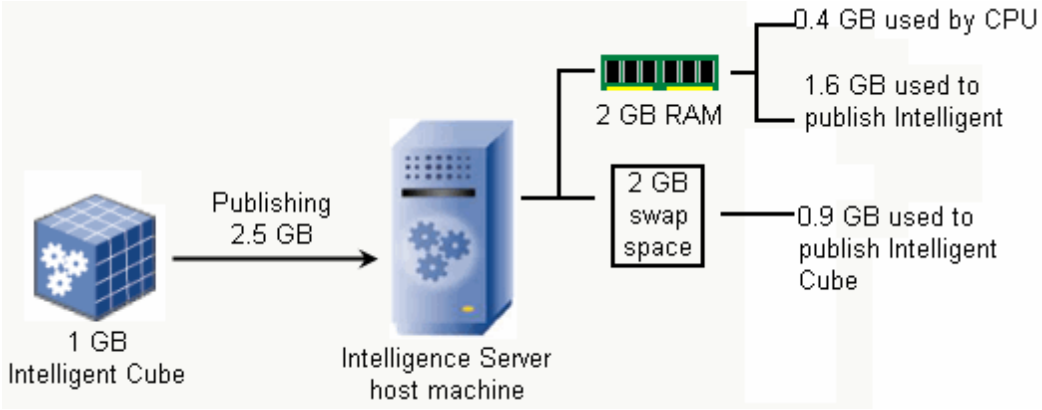
If publishing an Intelligent Cube is processed in RAM alone without using swap space, this can greatly reduce the effect publishing an Intelligent Cube has on the performance of your Intelligence Server host machine. Swap space is controlled by the operating system of a computer and using this space for the transfer of data into RAM can negatively affect the performance of a computer.

You can help to keep the processes of publishing Intelligent Cubes within RAM alone by defining memory limits for Intelligent Cubes that reflect your Intelligence Server host's available RAM as well as schedule the publishing of Intelligent Cubes at a time when RAM usage is low. For information on scheduling Intelligent Cube publishing, see the *MicroStrategy In-memory Analytics Guide*.

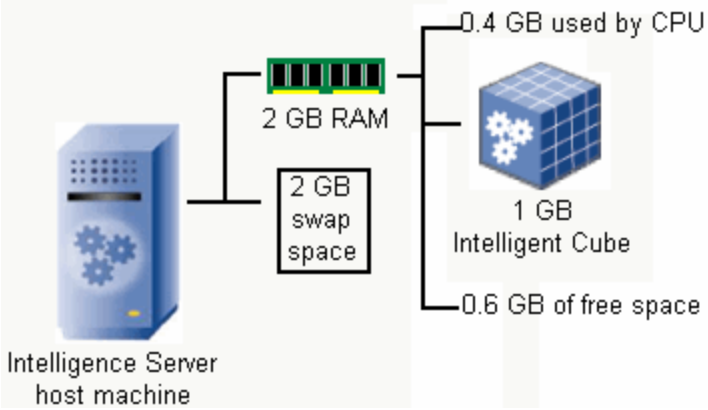
To determine memory limits for Intelligent Cubes, you should review the considerations listed in *Determining memory limits for Intelligent Cubes, page 465*. You must also account for the potential peak in memory usage when publishing an Intelligent Cube, which can be two to four times the size of an Intelligent Cube.

For example, your Intelligence Server machine has 2 GB of RAM and 2 GB of swap space. Assume that normal usage of RAM by the operating system uses .4 GB of RAM. This leaves a possible 1.6 GB of RAM for Intelligent Cube storage and other tasks.

With this configuration, consider the act of publishing a 1 GB Intelligent Cube. Assuming the peak in memory usage for publishing this Intelligent Cube is 2.5 times the size of the Intelligent Cube, the publishing process requires 2.5 GB. This can take up 1.6 GB of RAM, but it also requires .9 GB of swap space, as shown in the diagram below.



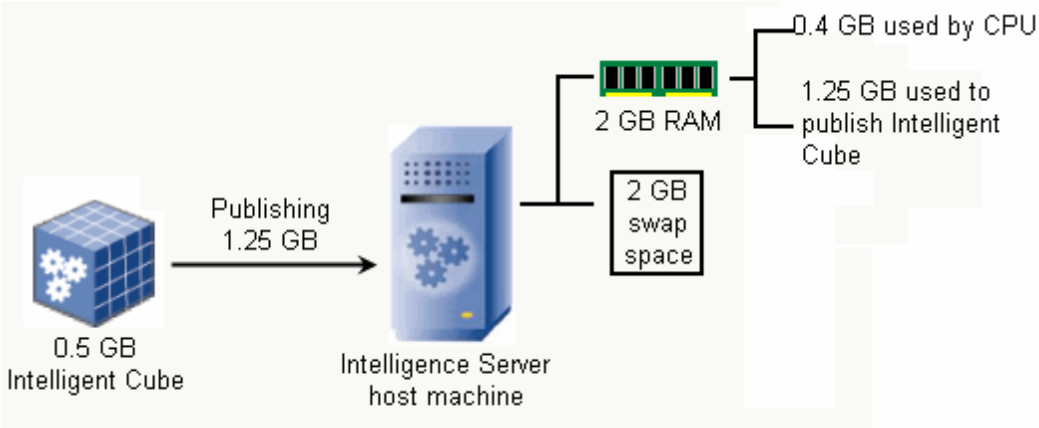
Once the Intelligent Cube is published, only the 1 GB for the Intelligent Cube (plus some space for indexing information) is used in RAM and the remaining .6 GB of RAM and .9 GB of swap space used during the publishing of the Intelligent Cube is returned to the system, as shown in the image below.



If Intelligence Server is hosted on an AIX machine, the resources required for Intelligent Cube publication may not be returned to the system. However, these resources can be used for additional Intelligence Server operations.

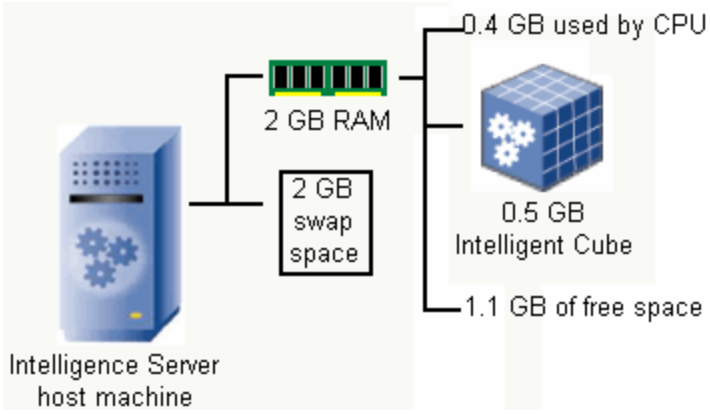
While the Intelligent Cube can be published successfully, using the swap space could have an affect on performance of the Intelligence Server machine.

With the same configuration, consider the act of publishing a .5 GB Intelligent Cube rather than a 1 GB Intelligent Cube. Assuming the peak in memory usage for publishing this Intelligent Cube is 2.5 times the size of the Intelligent Cube, the publishing process requires 1.25 GB. This process can be handled completely within RAM, as shown in the diagram below.



Once the Intelligent Cube is published, only the .5 GB for the Intelligent Cube (plus some space for indexing information) is used in RAM and the remaining RAM used during the publishing of the Intelligent Cube is returned to the system, as shown in the image below.

 © 2016, MicroStrategy Inc.



Be aware that as more Intelligent Cube data is stored in RAM, less RAM is available to process publishing an Intelligent Cube. This along with the peak memory usage of publishing an Intelligent Cube and the hardware resources of your Intelligence Server host machine should all be considered when defining memory limits for Intelligent Cube storage per project.

Defining memory limits for Intelligent Cubes

You can define limits for the amount of Intelligent Cube memory stored in Intelligence Server at a given time in two ways described below:

- You can use the amount of data required for all Intelligent Cubes to limit the amount of Intelligent Cube data stored in Intelligence Server memory at one time for a project. The default is 256 megabytes.

  The total amount of memory used on Intelligence Server by Intelligent Cubes for a project is calculated and compared to the limit you have defined. If an attempt to load an Intelligent Cube is made that will exceed this limit, an Intelligent Cube is removed from Intelligence Server memory before the new Intelligent Cube is loaded into memory.

- You can use the number of Intelligent Cubes to limit the number of Intelligent Cube stored in Intelligence Server memory at one time for a project. The default is 1000 Intelligent Cubes.

  The total number of Intelligent Cubes for a project that are stored in Intelligence Server memory is compared to the limit you define. If an attempt to load an Intelligent Cube is made that will exceed the numerical limit, an Intelligent Cube is removed from Intelligence Server memory before the new Intelligent Cube is loaded into memory.

To define limits on Intelligence Server memory usage by Intelligent Cubes

1. In Developer, log in to a project that uses Intelligent Cubes. You must log in using an account with the Administer Cubes privilege.

2 Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.

3 Expand **Intelligent Cubes**, and then select **General**.

4 Define the values for the following project governing options:

- **Maximum RAM usage (MBytes)**: Defines the data size, in megabytes, to limit the amount of Intelligent Cube data that can be stored in Intelligence Server memory for a project at one time. The default value is 256 megabytes.
- **Maximum number of cubes**: Defines the maximum total number of Intelligent Cubes that can be created for a project, including Intelligent Cubes that are not loaded into Intelligence Server's memory. The default value is 1000 Intelligent Cubes.

5 Click **OK** to save your changes and close the Project Configuration Editor.

Defining limits for Intelligent Cube indexes

Intelligence Server generates indexes to speed up access to data in Intelligent Cubes. In very large Intelligent Cubes, these indexes may significantly increase the size of the Intelligent Cube. You can define limits for how much the indexes can add to the size of the Intelligent Cube at the project level, using the Project Configuration Editor.

To define limits for the Intelligent Cube indexes

1 In Developer, log in to the project source for your project.

2 In the Folder List, right-click the project and choose **Project Configuration**. The Project Configuration Editor opens.

3 Expand **Intelligent Cubes**, and then select **General**.

4 Edit the following values, as appropriate:

- **Maximum % growth of an Intelligent Cube due to indexes**: Defines the maximum that indexes are allowed to add to the Intelligent Cube's size, as a percentage of the original size. For example, a setting of 50 percent defines that a 100 MB Intelligent Cube can grow to 150 MB because its indexes. If the Intelligent Cube's size exceeds this limit, the least-used indexes are dropped from the Intelligent Cube.
- **Cube growth check frequency (in mins)**: Defines, in minutes, how often the Intelligent Cube's size is checked, and if necessary, how often the least-used indexes are dropped.

5 Click **OK** to save the settings.

 © 2016, MicroStrategy Inc.

Defining limits for Intelligent Cubes created using the Import Data feature

The Import Data feature allows users to create Intelligent Cubes using external data sources, such as Excel files, tables in a database, or Freeform SQL queries. You can define limits on the size of files that users can upload, and on the total memory available for the Intelligent Cubes they can create.

To define limits on Intelligent Cubes created using the Import Data feature

1. In Developer, log in to a project using an account with the Administer Cubes privilege.
2. In the Folder List, right-click on the project and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Governing Rules**, expand **Default**, and then select **Import Data**.
4. Define values for the following options:
   - **Maximum file size (MB)**: Defines the maximum size for files that users can upload and import data from. The default value is 30 MB. The minimum value is 1 MB, and the maximum value is 500 MB.
   - **Maximum quota per user (MB)**: Defines the maximum size of all data import cubes for each individual user. This quota includes the file size of all data import cubes, regardless of whether they are published to memory or on disk. The default value is 100 MB, and the minimum value is 0.

     In a clustered environment, this setting applies to all nodes in the cluster.
5. Click **OK** to save your changes and close the Project Configuration Editor.

Loading Intelligent Cubes when Intelligence Server starts

When Intelligence Server is started there are various tasks that are processed to prepare a reporting environment. You can include loading all published Intelligent Cubes as one of the tasks completed when Intelligence Server is started. This affects when the load time required for Intelligent Cubes occurs.

The considerations for whether to load Intelligent Cubes at Intelligence Server startup or when a report is executed that accesses a published Intelligent Cube are described in the table below:

Method	Pros	Cons
Loading Intelligent Cubes when Intelligence Server starts	• Report runtime performance for reports accessing Intelligent Cubes is optimized since the Intelligent Cube for the report has already been loaded. • This practice is a good option if Intelligent Cubes are commonly used in a project.	• The overhead experienced during Intelligence Server startup is increased due to the processing of loading Intelligent Cubes. • All Intelligent Cubes for a project are loaded into Intelligence Server memory, regardless of whether they are used by reports or not.
Loading Intelligent Cubes when a report is executed that accesses a published Intelligent Cube	• The overhead experienced during Intelligence Server startup is decreased as compared to including loading Intelligent Cubes as part of the startup tasks. • If Intelligent Cubes are not required by any reports, then they do not need to be loaded into Intelligence Server and no overhead is experienced. • This practice is a good option if Intelligent Cubes are supported for a project, but some of the Intelligent Cubes are rarely used in the project.	• Report runtime performance for reports accessing Intelligent Cubes can be negatively affected as the Intelligent Cube must first be loaded into Intelligence Server. You can also load Intelligent Cubes manually or with subscriptions after Intelligence Server is started.

The procedure below describes how to enable or disable loading Intelligent Cubes when Intelligence Server starts.

The act of loading an Intelligent Cube can require memory resources in the area of two times greater than the size of an Intelligent Cube. This can affect performance of your Intelligence Server as well as the ability to load the Intelligent Cube. For information on how to plan for these memory requirements, see *Planning for Intelligent Cube publishing and loading , page 466*.

To enable or disable loading Intelligent Cubes when Intelligence Server starts

1. In Developer, log in to a project with a user account with administrative privileges.
2. Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Intelligent Cubes**, and then select **General**.
4. Select or clear the **Load Intelligent cubes on startup** check box to enable or disable loading Intelligent Cubes when Intelligence Server starts.
5. Click **OK** to save your changes and close the Project Configuration Editor.

 © 2016, MicroStrategy Inc.

Storing Intelligent Cubes in secondary storage

Along with storing Intelligent Cubes in Intelligence Server memory, you can also store them in secondary storage, such as a hard disk. These Intelligent Cubes can then be loaded from secondary storage into Intelligence Server memory when reports require access to the Intelligent Cube data.

To store an Intelligent Cube in secondary storage

1 In Developer, log in to a project source with administrative privileges.

To modify the secondary storage location of Intelligent Cubes

Before you save Intelligent Cubes to secondary storage, use the following steps to define where Intelligent Cubes are saved.

2 Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.

3 Expand **Intelligent Cubes**, and then select **General**.

4 In the **Intelligent Cube file directory** area, click **...** (the **Browse** button). The Browse for Folder dialog box opens.

5 Browse to the folder location to store Intelligent Cubes, and then click **OK**. You are returned to the Project Configuration Editor.

6 Click **OK** to save your changes and close the Project Configuration Editor.

To save an Intelligent Cube to secondary storage

7 From the **Folder List**, expand **Administration**, then expand **System Monitors**, then expand **Caches**, and then select **Intelligent Cubes**. The Intelligent Cube Monitor is displayed.

8 Right-click the Intelligent Cube to store in secondary storage and select **Save to Disk**. The Intelligent Cube is saved to the secondary storage location.

You can also define when Intelligent Cubes are automatically saved to secondary storage, as described in *Defining when Intelligent Cubes are automatically saved to secondary storage, page 473* below.

Defining when Intelligent Cubes are automatically saved to secondary storage

In addition to manually saving Intelligent Cubes to secondary storage, you can also define when Intelligent Cubes are automatically saved to secondary storage.

To define when Intelligent Cubes are automatically saved to secondary storage

1. In Developer, log in to a project source with administrative privileges.
2. From the **Administration** menu, point to **Server**, and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.
3. Expand the **Server Definition** category, and select **Advanced**.
4. In the **Backup frequency (minutes)** field, type the interval (in minutes) between when Intelligent Cubes are automatically saved to secondary storage. A value of 0 means that Intelligent Cubes are backed up immediately after they are created or updated.

   Be aware that this option also controls the frequency at which cache and History List messages are backed up to disk, as described in *Backup Frequency (minutes), page 417*.
5. Click **OK** to save your changes and close the Intelligence Server Configuration Editor.
6. Restart Intelligence Server for your changes to take effect.

Supporting connection mappings in Intelligent Cubes

Connection mappings allow you to assign a user or group in the MicroStrategy system to a specific login ID on the data warehouse. Connection mappings are typically used for one of the following reasons:

- To take advantage of one of several RDBMS data security techniques (security views, split fact tables by rows, split fact tables by columns) that you may have already created
- To allow users to connect to multiple data warehouses using the same project

For detailed information about connection mapping, see the *MicroStrategy Installation and Configuration Guide*.

If you use connection mapping in a project that includes Intelligent Cubes, you should define your Intelligent Cubes to use and support connection mapping. If you do not define Intelligent Cubes to support connection mapping when connection mapping is used in a project, users may be able to access data they are not intended to have access to.

When an Intelligent Cube that supports connection mapping is published, it uses the connection mapping of the user account which published the Intelligent Cube. Only

 © 2016, MicroStrategy Inc.

users that have this connection mapping can create and view reports that access this Intelligent Cube. This maintains the data access security and control defined by your connection mappings.

If an Intelligent Cube needs to be available for multiple connection mappings, you must publish a separate version of the Intelligent Cube for each of the required connection mappings.

For example, Intelligent Cube X is created in a project and defined to support connection mapping. User JDoe who is assigned to connection mapping A publishes Intelligent Cube X. The Intelligent Cube is published using connection mapping X. User FJohnson who is assigned connection mapping B cannot create and execute a report connected to Intelligent Cube X. To allow FJohnson to create and execute a report connected to Intelligent Cube X, a user account assigned to connection mapping B must publish the Intelligent Cube.

To support connection mapping for all Intelligent Cubes in a project

1. In Developer, log in to a project with a user account with administrative privileges.
2. Right-click a project and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand **Intelligent Cubes**, and then select **General**.
4. Select the **Create Intelligent Cubes by database connection** check box.

   If you do not use connection mapping, leave this check box cleared.
5. Click **OK**. The Project Configuration Editor closes.

 © 2016, MicroStrategy Inc.

12

SCHEDULING JOBS AND ADMINISTRATIVE TASKS

Introduction

Scheduling is a feature of Intelligence Server that you can use to automate various tasks. Time-sensitive, time-consuming, repetitive, and bulk tasks are ideal candidates for scheduling. Running a report or document is the most commonly scheduled task because scheduling reports, in conjunction with other features such as caching and clustering, can improve the overall system performance. Certain administration tasks can also be scheduled.

Intelligence Server executes a task in exactly the same manner if it is scheduled or not. All governing parameters and error conditions apply to scheduled tasks just as they apply to other requests.

The scheduling feature is turned on by default. However, you can disable scheduling in the Intelligence Server Configuration Editor. In the **Server Definition** category, in the **Advanced** subcategory, clear the **Use MicroStrategy Scheduler** check box.

This section discusses the following scheduling items:

- *Best practices for scheduling jobs and administrative tasks, page 478*
- *Creating and managing schedules, page 479*
- *Scheduling administrative tasks, page 485*
- *Scheduling reports and documents: Subscriptions, page 488*

© 2016, MicroStrategy Inc.

- *Configuring and administering Distribution Services, page 501*

Best practices for scheduling jobs and administrative tasks

MicroStrategy recommends the following best practices when scheduling jobs and administrative tasks:

- Executing simultaneous reports can strain system resources. If you have many reports or tasks that need to be executed on the same time-based schedule, consider creating several similar schedules that trigger 15 minutes apart. For example, one schedule triggers at 8 AM every Monday, and another triggers at 8:15 AM.
- To prevent users from scheduling many simultaneous reports, you can prevent users from scheduling jobs using a schedule by editing the schedule's Access Control List (ACL). To do this, in the Schedule Manager, right-click the schedule and select **Properties**, then select the Security tab in the Properties dialog box, and make sure that only users who can use the schedule have Modify or Full Control access to the schedule. For more information about ACLs, see *Controlling access to objects: Permissions, page 58*.
- Establish reasonable limits on how many scheduled jobs are allowed. For details on this setting, see *Limiting the total number of jobs, page 332*.
- If you need to create multiple similar subscriptions, you can create them all at once with the Subscription Wizard. For example, you can subscribe users to several reports at the same time.
- If you need to temporarily disable a schedule, you can set its start date for some time in the future. The schedule does not trigger any deliveries until its scheduled start date.
- In a clustered system, if it is important which node an administrative task is executed on, use an event-triggered schedule and trigger the event on that node.
- If many subscriptions are listed in the Subscription Manager, you can filter the list of subscriptions so that you see the relevant subscriptions. For more information, including instructions, see the *MicroStrategy Developer Help*.
- When selecting reports to be subscribed to, make sure all the reports with prompts that require an answer actually have a default answer. If a report has a prompt that requires an answer but has no default answer, the subscription cannot run the report successfully because the prompt cannot be resolved, and the subscription is automatically invalidated and removed from the system.
- When a scheduled report or document finishes executing, a message can display in the subscribed user's History List alerting her that the report is ready to be viewed. The user then opens the message to retrieve the results. If the request was not completed successfully, the user can view details of the error message. These

© 2016, MicroStrategy Inc.

messages are available in the History List folder. For more information about History Lists, see *Saving report results: History List, page 425*.

- You can track the successful delivery of a subscribed report or document. In the Subscription Editor or Subscription Wizard, select the **Send notification to email address** check box and specify the email address. A notification email is sent to the selected address when the subscribed report or document is successfully delivered to the recipients.
- You can track the failed delivery of subscribed reports or documents. In the Project Configuration Editor, in the **Deliveries: Email notification** category, enable the administrator notification settings for failed deliveries.
- For best performance, consider configuring the following settings to suit your subscription needs:
  - Tune the **Number of scheduled jobs** governing setting according to the size of your hardware. Larger hardware can handle higher settings.
  - Enable caching.
  - If your database and database machine allow a larger number of warehouse connections, increasing this number can improve performance by allowing more jobs to execute against the warehouse.
  - Increase the **Scheduler session timeout** setting.

  It is strongly recommended that you exercise caution when changing settings from the default. For details on each setting, see the appropriate section of this manual.
- To control memory usage, consider configuring the following settings:
  - Limit the number of scheduled jobs per project and per Intelligence Server.
  - Increase the User session idle time.
  - Enable caching.
- If you are using Distribution Services, follow the best practices provided in *Best practices for using Distribution Services, page 504*.

Creating and managing schedules

A *schedule* is a MicroStrategy object that contains information specifying when a task is to be executed. One schedule can control several tasks. Schedules are stored at the project source level and are thus available to all projects in the project source.

Intelligence Server supports two types of schedules:

© 2016, MicroStrategy Inc.

- Time-triggered schedules execute at a date and time, or on a recurring date and time. For details, see *Time-triggered schedules, page 480*.
- Event-triggered schedules execute when the event associated with them is triggered. For details, see *Event-triggered schedules, page 480*.

Time-triggered schedules

With a time-triggered schedule, you define a date and time at which the scheduled task is to be run. For example, you can execute a task every Sunday night at midnight. Time-triggered schedules are useful to allow large, resource-intensive tasks to run at off-peak times, such as overnight or over a weekend.

Note the following:

- Time-triggered schedules execute according to the time on the machine where they were created. For example, a schedule is created using Developer on a machine that is in the Pacific time zone (GMT -8:00). The schedule is set to be triggered at 9:00 AM. The machine is connected to an Intelligence Server in the Eastern time zone (GMT -5:00). The schedule executes at 12:00 PM Eastern time, which is 9:00 AM Pacific time.
- In a clustered environment, administrative tasks associated with time-triggered schedules are executed on only the primary node of the cluster.

Event-triggered schedules

An event-triggered schedule causes tasks to occur when an event occurs. For example, an event may trigger when the database is loaded, or when the books are closed at the end of a cycle.

When an event is triggered, all tasks tied to that event through an event-triggered schedule begin processing. For more information about events, including how to create them, see *About events and event-triggered schedules, page 483*.

In a clustered environment, administrative tasks associated with event-triggered schedules are executed on only the node of the cluster that triggered the event.

Creating schedules

To create schedules, you must have the privileges Create Configuration Object and Create and Edit Schedules and Events. In addition, you need to have Write access to the Schedule folder. For information about privileges and permissions, see *Controlling access to application functionality, page 58*.

To create effective and useful schedules, you must have a clear understanding of your users' needs and the usage patterns of the overall system. Schedules must be created before they are linked to any tasks.

 © 2016, MicroStrategy Inc.

To create a schedule

1. In Developer, log in to a project source.
2. Expand **Administration**, then expand **Configuration Managers**, and then select **Schedules**. The list of schedules for the project source displays on the right-hand side.
3. From the **File** menu, point to **New**, and then select **Schedule**. The Schedule Wizard opens.
4. Step through the wizard, entering the required information:
   - To create a time-triggered schedule, when prompted for the schedule type, select **Time-triggered**. Then select the frequency and time the schedule is triggered.
   - To create an event-triggered schedule, when prompted for the schedule type, select **Event-triggered**. Then select the event that triggers the schedule.
5. When you reach the Summary page of the wizard, review your choices and click **Finish**. The schedule is created.

You can also create a schedule with the Create Schedule script for Command Manager. For detailed syntax, see the Create Schedule script outline in Command Manager.

Managing schedules

You can add, remove, or modify schedules through the Schedule Manager. You can modify the events that trigger event-triggered schedules through the Event Manager. For instructions on using the Event Manager, see *About events and event-triggered schedules, page 483*.

You can also specify that certain schedules can execute subscriptions relating only to certain projects. For instructions, see *Restricting schedules, page 482*.

To manage your schedules in the Schedule Manager

1. In Developer, log in to a project source.
2. Expand **Administration**, then expand **Configuration Managers**, and then select **Schedules**. The list of schedules for the project source displays on the right-hand side.
3. To manage your schedules, select from the tasks below:
   - To create a new schedule, see *Creating schedules, page 480*.

© 2016, MicroStrategy Inc.

- To modify a schedule, right-click the schedule and select **Edit**. The Schedule Wizard opens with that schedule's information. Step through the wizard and make any changes.
- To delete a schedule, right-click the schedule and select **Delete**. The schedule is deleted.
- To find all subscriptions that use one of the schedules, right-click the schedule and select **Search for dependent subscriptions**. A list of subscriptions that use that schedule opens.

Restricting schedules

You may want to restrict some schedules so that they can be used only by subscriptions in specific projects. For example, your On Sales Database Load schedule may not be relevant to your Human Resources project. You can configure the Human Resources project so that the On Sales Database Load schedule is not listed as an option for subscriptions in that project.

You may also want to restrict schedules so that they cannot be used to subscribe to certain reports. For example, your very large All Worldwide Sales Data document should not be subscribed to using the Every Morning schedule. You can configure the All Worldwide Sales Data document so that the Every Morning schedule is not listed as an option for subscriptions to that document.

To restrict schedules for a project

1. In MicroStrategy Web, log in to the project you are restricting schedules for. You must log in as a user with administrative access to the MicroStrategy Web preferences.
2. Click the **MicroStrategy** icon, then click **Preferences**. The Web Preferences page opens.
3. In the left column, click **Project Defaults**, and then click **Schedule**. The Schedule preferences are displayed.
4. Select **Only allow users to subscribe to the schedules below**.
5. The left column lists schedules that users are not allowed to subscribe to. The right column lists schedules that users are allowed to subscribe to.

When you first select this option, no schedules are allowed. All schedules are listed by default in the left column, and the right column is empty.

6. To allow users to subscribe to a schedule, select the schedule and click the right arrow. The schedule moves to the right column.
7. When you are finished selecting the schedules that users are allowed to subscribe to in this project, click **Save**. The list of schedules is saved and you can leave the MicroStrategy Web Preferences page.

 © 2016, MicroStrategy Inc.

To restrict schedules for a report or document

1. In MicroStrategy Web, log in to the project you are restricting schedules for. You must log in as a user with administrative access to the MicroStrategy Web preferences.
2. Execute the report or document.
3. From the **Tools** menu, select **Report Options**. The Report Options dialog box opens.
4. Select the **Advanced** tab.
5. Select **Only allow users to subscribe to the schedules below**.
6. The left column lists schedules that users are not allowed to subscribe to. The right column lists schedules that users are allowed to subscribe to.

   When you first select this option, no schedules are allowed. All schedules are listed by default in the left column, and the right column is empty.

7. To allow users to subscribe to a schedule, select the schedule and click the right arrow. The schedule moves to the right column.
8. When you are finished selecting the schedules that users are allowed to subscribe to in this project, click **OK**. The list of schedules is saved, and the Report Options dialog box closes.

About events and event-triggered schedules

Subscriptions and tasks that are based on event-triggered schedules (see *Event-triggered schedules, page 480*) execute when a MicroStrategy event is triggered. These triggers do not need to be defined in advance. A system external to Intelligence Server is responsible for determining whether the conditions for triggering an event are met. For more information on how to trigger events, see *Triggering events, page 484*.

Once Intelligence Server has been notified that the event has taken place, Intelligence Server performs the tasks associated with the corresponding schedule.

In a clustered environment, administrative tasks associated with event-triggered schedules are executed only by the node on which the event is triggered. MicroStrategy recommends that you use event-triggered schedules in situations where it is important to control which node performs certain tasks.

If projects are distributed asymmetrically across the cluster, when you assign an event-triggered schedule to a project, make sure you trigger the event on all nodes on which that project is loaded. See *Scheduling administrative tasks in a clustered system, page 487*.

© 2016, MicroStrategy Inc.

Creating events

You can create events in Developer using the Event Manager.

To create an event in Developer

1. In Developer, log in to a project source. You must log in as a user with the Create And Edit Schedules And Events privilege.
2. Expand **Administration**, then expand **Configuration Managers**, and then select **Events**. The list of events for the project source displays on the right-hand side.
3. From the **File** menu, point to **New**, and then select **Event**. A new event is created.
4. Name the new event.

To create an event using Command Manager

You can create events with the following Command Manager script:

```
CREATE EVENT event_name [DESCRIPTION description];
```

By default, this script is in the folder `C:\Program Files\MicroStrategy\Command Manager\Outlines\`.

Triggering events

MicroStrategy Command Manager can trigger events from the Windows command line. By executing Command Manager scripts, external systems can trigger events and cause the associated tasks to be run. For more information about Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*.

For example, you want to execute several reports immediately after a database load occurs so that these reports always have a valid cache available. You create an event called OnDBLoad and associate it with an event-triggered schedule. You then subscribe those reports to that schedule.

At the end of the database load routine, you include a statement to add a line to a database table, DB_LOAD_COMPLETE, that indicates that the database load is complete. You then create a database trigger that checks to see when the DB_LOAD_COMPLETE table is updated, and then executes a Command Manager script. That script contains the following line:

```
TRIGGER EVENT "OnDBLoad";
```

When the script is executed, the OnDBLoad event is triggered, and the schedule is executed.

 © 2016, MicroStrategy Inc.

You can also use the MicroStrategy SDK to develop an application that triggers an event. You can then cause the database trigger to execute this application.

Triggering events manually

You can manually trigger events using the Event Manager. This is primarily useful in a testing environment. In a production system, it may not be practical for the administrator to be present to trigger event-based schedules.

To trigger an event manually

1. In Developer, log in to a project source. You must log in as a user with the Trigger Event privilege.
2. Expand **Administration**, then expand **Configuration Managers**, and then select **Events**. The list of events for the project source displays on the right-hand side.
3. Right-click an event and select **Trigger**. The event is triggered.

Scheduling administrative tasks

In addition to scheduling reports and documents execution, you can instruct Intelligence Server to perform certain administrative tasks according to a schedule. For example, you can delete all History List messages every month, or idle a project once a week for maintenance and then resume it an hour later.

To schedule an administrative task, you must have the Administer Subscriptions privilege and any privileges required for that task.

To schedule an administrative task

1. In Developer, from the **Administration** menu, point to **Scheduling** and then select **Schedule Administration Tasks**. The Schedule Administration Tasks dialog box opens.
2. To schedule tasks for a project, select that project. To schedule tasks for the project source, select All Projects.
3. Choose a task from the action list. For descriptions of the tasks, see the table below.
4. Select one or more schedules for the task.
5. Set any additional options required for the task. For information about the possible options for each task, click **Help** in the Schedule Administration Tasks dialog box.

© 2016, MicroStrategy Inc.

6 Click **OK**. The Schedule Administration Tasks dialog box closes and the task is scheduled.

The table below lists the tasks that can be scheduled for a project. Some of the tasks can also be scheduled at the project source level, affecting all projects in that project source.

Task	Description
Cache or History List management tasks	
Delete report caches	Delete all report caches for the project. For more information, see *Deleting result caches, page 414*. **Note**: Typically the Invalidate Caches task is sufficient to clear the report caches.
Clean History List database	Delete orphaned entries and ownerless inbox messages from the History List database. For more information, see *Cleaning up the History List Database, page 438*
Delete History List messages (project or project source)	Delete all history list messages for the project or project source. For more information, see *Scheduling History List message deletion, page 437*. **Note**: This maintenance request can be large. Schedule the History List deletions for times when Intelligence Server is not busy, such as when users are not sending requests to the system. Alternatively, delete History Lists in increments; for example, delete the History Lists of groups of users at different times, such as at 1 AM, 2 AM, and so on.
Invalidate caches	Invalidate the report caches in a project. The invalid caches are automatically deleted once all references to them have been deleted. For more information, see *Invalidating result caches, page 413*.
Purge element caches	Delete the element caches for a project. For more information, see *Deleting all element caches, page 450*.
Intelligent Cube management tasks	
Activate Intelligent Cubes	Publish an Intelligent Cube to Intelligence Server, making it available for use in reports. For more information, see *Chapter 11, Managing Intelligent Cubes*.
Deactivate Intelligent Cubes	Unpublish an Intelligent Cube from Intelligence Server. For more information, see *Chapter 11, Managing Intelligent Cubes*.
Delete Intelligent Cube	Delete an Intelligent Cube from the server. For more information, see *Chapter 11, Managing Intelligent Cubes*.
Update Intelligent Cubes	Update a published Intelligent Cube. For more information, see *Chapter 11, Managing Intelligent Cubes*.
Project management tasks	
Idle project	Cause the project to stop accepting certain types of requests. For more information, see *Setting the status of a project, page 27*.
Load project	Bring the project back into normal operation from an unloaded state. For more information, see *Setting the status of a project, page 27*.
Resume project	Bring the project back into normal operation from an idle state. For more information, see *Setting the status of a project, page 27*.

 © 2016, MicroStrategy Inc.

Task	Description
Unload project	Take a project offline to users and remove the project from Intelligence Server memory. For more information, see *Setting the status of a project, page 27*.
Miscellaneous management tasks	
Batch LDAP import (project source only)	Import LDAP users into the MicroStrategy system. For more information, see *Importing LDAP users and groups into MicroStrategy, page 127*.
Delete unused managed objects (project or project source)	Remove the unused managed objects created for Freeform SQL, Query Builder, and MDX cube reports. For more information, see *Deleting unused schema objects: managed objects, page 276*.
Deliver APNS Push Notification	Deliver a push notification for a Newsstand subscription to a mobile device. For more information, see the *MicroStrategy Mobile Design and Administration Guide*.

Managing scheduled administration tasks

The Scheduled Maintenance view of the System Administration monitor lists all the scheduled administrative tasks for a project source. From this view you can see information about all the scheduled tasks or delete one or more tasks. For more information about using the System Administration monitor, see *Managing and monitoring projects, page 24*.

To manage scheduled administration tasks

1. In Developer, log in to a project source. You must log in as a user with the Administer Subscriptions privilege.
2. Expand **Administration**, and then expand **System Administration**.
3. Select **Scheduled Maintenance**. The list of administration tasks that are scheduled for the project source displays in the right-hand side.
4. To view detailed information about a scheduled task, right-click the task and select **Quick View**.
5. To delete a scheduled task, right-click the task and select **Expire**. The task is removed from the list of tasks.

Users are not notified when a task they have scheduled is deleted.

Scheduling administrative tasks in a clustered system

When you set up several Intelligence Server machines in a cluster, you can distribute projects across those clustered machines (or nodes) in any configuration. Each node can

host a different subset of projects. For more information about clustering Intelligence Servers, see *Chapter 9, Clustering Multiple MicroStrategy Servers*.

To determine which server handles each scheduled administrative task, use the following guidelines:

- Tasks that are based on time-based schedules are executed on the primary node for each project. You can find a project's primary node using the Cluster view of the System Administration monitor.
- Tasks that are based on event-triggered schedules are executed on the node on which the event is triggered. The administrator must be sure to trigger the event on all nodes (that is, all machines) that are running the project for which the event-based schedule is assigned.

You can see which nodes are running which projects using the Cluster view of the System Administration monitor. For details on using the Cluster view of the System Administration monitor, see *Managing your clustered system, page 390*.

Scheduling reports and documents: Subscriptions

Normally, Intelligence Server executes report or document requests immediately after they are made. A subscription allows these requests to be executed according to a schedule specified by the administrator. Users can create subscriptions for themselves, or system administrators can subscribe users to reports. In addition, if you have a Distribution Services license, you can deliver subscribed reports or documents to other users by email, file, FTP, or print.

Scheduling reports and documents execution reduces the load on the system in the following ways:

- You can create caches for frequently accessed reports and documents, which provides fast response times to users without generating additional load on the database system.
- Large, long-running reports and documents can be postponed to later when the system load is lighter.

A subscription for a document creates or updates only that document's cache for the default mode of the document (HTML, PDF, Excel, or XML/Flash). If the document is viewed in other modes, it does not use this cache. For more information about how Intelligence Server determines whether to use a cache, see *Cache matching algorithm, page 406*.

When a user subscribes or is subscribed to a report or document, that user's personalization selections apply to the subscription. Personalization selections can include language choice, delivery method, delivery location, delivery format, and so on.

 © 2016, MicroStrategy Inc.

Personalization options vary depending on what a report or document supports, whether the user's MicroStrategy environment is internationalized in the appropriate language for the user, and so on.

This section provides the following information about subscriptions:

- *Types of subscriptions, page 489*
- *Creating subscriptions, page 490*
- *Managing subscriptions, page 498*

Types of subscriptions

You can create the following types of subscriptions for a report or document:

- **Cache update** subscriptions refresh the cache for the specified report or document. For example, your system contains a set of standard weekly and monthly reports. These reports should be kept in cache because they are frequently accessed. Certain tables in the database are refreshed weekly, and other tables are refreshed monthly. Whenever these tables are updated, the appropriate caches should be refreshed.

  Cache update subscriptions often use event-triggered schedules because caches generally do not need to be refreshed unless the underlying data changes from an event like a data warehouse load. For additional suggestions for scheduling strategies, see *Scheduling updates of result caches, page 412*. For detailed information about caches, see *Result caches, page 400*.

  WebDAV subscriptions are a special type of cache update subscriptions. These subscriptions update an Intelligence Server folder whose contents are hosted by a WebDAV server so that the information in the folder can be served to mobile devices. For information about WebDAV folders, see the *Advanced Reporting Guide*.

- **History List** subscriptions create a History List message for the specified report or document. Users can then retrieve the report or document from the History List message in their History List folder. For detailed information about the History List, see *Saving report results: History List, page 425*.

- **Mobile** subscriptions deliver the report or document to a mobile device, such as an iPhone or an Android device, via MicroStrategy Mobile. These subscriptions are available if you own the MicroStrategy Mobile product. For detailed information about mobile subscriptions and MicroStrategy Mobile, see the *MicroStrategy Mobile Design and Administration Guide*.

- **Intelligent Cube update** subscriptions retrieve the most recent information for an Intelligent Cube from the data warehouse and then publish that Intelligent Cube. Like cache update subscriptions, Intelligent Cube update subscriptions are good candidates for event-triggered schedules. For detailed information about Intelligent Cubes, see the *MicroStrategy In-memory Analytics Guide*.

- **Email** subscriptions deliver a report or document to one or more email addresses.

- **File** subscriptions save the report or document as an Excel or PDF file to a disk location on the network.

© 2016, MicroStrategy Inc.

- **Print** subscriptions automatically print a report or document from a specified printer.
- **FTP** subscriptions automatically save the report or document to a location on an FTP server in the file format the user chooses: command separated values (CSV), PDF, HTML, MS Excel, and plain text.

Email, file, print, and FTP subscriptions are available if you have purchased a Distribution Services license. For information on purchasing Distribution Services, contact your MicroStrategy account representative.

Distribution Services subscriptions

If you have a Distribution Services license, you can set up information flows for yourself and other users by subscribing to report and document deliveries. Users can freely personalize these deliveries by selecting delivery formats and locations, such as:

- Format: HTML, Flash, PDF, Excel, ZIP file, plain text, CSV, or bulk export
- Delivery location: Email, network printer, FTP location, file server (including portals and PCs), or the user's MicroStrategy History List, which serves as a report archive and immediately informs the user of the delivery by email

Reports or documents that are subscribed to for delivery through Distribution Services can be compressed and password protected. Standard MicroStrategy security credentials are applied for each user subscribed to receive a report or document.

Before you can use Distribution Services to deliver reports and documents, you must first create the appropriate devices, transmitters, and contacts. For detailed information on these objects and instructions on setting up your Distribution Services system, see *Configuring and administering Distribution Services, page 501*.

Creating subscriptions

This section provides detailed instructions for subscribing to a report or document. It also provides information on the following:

- *Prompted reports and subscriptions, page 493*
- *Using a report to specify the recipients of a subscription, page 493*
- *Personalizing email and file subscriptions, page 495*
- *Delivering parts of reports across multiple files: Bursting file subscriptions, page 496*
- *Delivering parts of reports across multiple files: Bursting file subscriptions to subfolders, page 497*

You can create subscriptions in the following ways:

- You can subscribe to an individual report or document from the Report Editor or Document Editor in Developer or through MicroStrategy Web (see *To subscribe to a*

 © 2016, MicroStrategy Inc.

report or document in Developer, page 492 or *To create a subscription in MicroStrategy Web, page 492*).

Note the following:

Use this method to create WebDAV folder update subscriptions.

A History List message is generated when a report or document is executed in Web by a schedule.

- If you have a Distribution Services license, you can subscribe multiple users to an individual report or document through MicroStrategy Web (see *To create a subscription in MicroStrategy Web, page 492*).
- You can create multiple cache, History List, Intelligent Cube, or Mobile subscriptions at one time for a user or user group using the Subscription Wizard in Developer (see *To create multiple subscriptions at one time in Developer, page 492*).
- If you have purchased a license for Command Manager, you can use Command Manager scripts to create and manage your schedules and subscriptions. For instructions on creating these scripts with Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*, or see the Command Manager Help. (From within Command Manager, select **Help**.)

Prerequisites

- To create any subscriptions, you must have the Schedule Request privilege.
- To create email, file, FTP, or print subscriptions, you must have a MicroStrategy Web license, a Distribution Services license, and the appropriate privileges in the Distribution Services privilege group. For example, to create an email subscription you must have the Use Distribution Services and Subscribe to Email privileges.

  To create an alert-based subscription, you must also have the Web Create Alert privilege (under the Web Analyst privilege group).
- To create mobile subscriptions, you must have a MicroStrategy Mobile license.
- To subscribe other users to a report or document, you must have the Web Subscribe Others privilege (under the Web Professional group). In addition, to subscribe others in Developer, you must have the Administer Subscriptions, Configure Subscription Settings, and Monitor Subscriptions privileges (under the Administration group).
- To subscribe a dynamic address list to a report or document, you must have the Subscribe Dynamic Address List privilege. For information about dynamic address lists, see *Using a report to specify the recipients of a subscription, page 493*.

© 2016, MicroStrategy Inc.

To subscribe to a report or document in Developer

Only History List, cache, Intelligent Cube, and Mobile subscriptions can be created in Developer.

1. In Developer, select the report, document, Intelligent Cube, or WebDAV folder to be delivered according to a schedule.
2. From the **File** menu, point to **Schedule Delivery To**, and select the type of subscription to create. The Subscription Editor for that type of subscription opens. For a list of the types of subscriptions, see *Types of subscriptions, page 489*. For detailed instructions on using the Subscription Editor, click **Help**.
3. Type a name and description for the subscription.
4. From the **Schedule** drop-down list, select a schedule for the subscription.
5. Click **OK** to save and close the Subscription Editor. The subscription is created and the report or document is delivered, or the Intelligent Cube is updated, according to the schedule.

To create multiple subscriptions at one time in Developer

Only History List, cache, Intelligent Cube, and Mobile subscriptions can be created in Developer.

1. In Developer, from the **Administration** menu, point to **Scheduling**, and then select **Subscription Creation Wizard**. The Subscription Wizard opens.
2. Step through the wizard, specifying a schedule and type for the subscriptions, and the reports and documents that are subscribed to. For detailed instructions, click **Help** on any page of the wizard.
3. When you have reached the Summary page of the wizard, review the subscription information and click **Finish**. The subscription is created and available for viewing in the Subscription Manager.

To create a subscription in MicroStrategy Web

1. In MicroStrategy Web, on the reports page, under the name of the report/document that you want to create a subscription for, click the **Subscriptions** icon .

   This icon becomes visible when you point to the name of the report or document.

 © 2016, MicroStrategy Inc.

2. Select **Add Subscription** for the type of subscription you want to create. For a list of the types of subscriptions, see *Types of subscriptions, page 489*. For detailed instructions on creating a subscription, click **Help**.
3. Type a name and description for the subscription.
4. From the **Schedule** drop-down list, select a schedule for the subscription.
5. To add additional users to the subscription, click **To**. The Recipient Browser dialog box opens. Select the users or groups and click **OK**. The Recipient Browser dialog box closes and the users and groups are added to the recipient list.
6. Click **OK** to save the subscription. The subscription is created and the report or document is delivered, or the Intelligent Cube is updated, according to the schedule.

Prompted reports and subscriptions

A subscribed report can contain prompts. How and whether the report is executed depends on the prompt definition. For additional information about how prompts are defined, see the *Prompts* chapter in the *Advanced Reporting Guide*.

To ensure that a prompted report in a subscription is executed properly, the prompt must be required and must have either a default answer or a personalized answer. The following table explains how Intelligence Server resolves the different possible scenarios that can occur for each prompt in a subscribed report.

Prompt Required?	Default / Personal Answer present?	Result
No	No	The prompt is ignored because it is not required; the report is executed, but it is not filtered by the prompt.
No	Yes	The prompt and its default or personal answer are ignored because the prompt is not required; the report is executed, but it is not filtered by the prompt.
Yes	No	The report is not executed. No answer is provided to the required prompt so MicroStrategy cannot complete the report without user interaction.
Yes	Yes	The report is executed. The prompt is answered with a personal answer if one is available or with the default answer if a personal answer is not provided.

Using a report to specify the recipients of a subscription

If you have a Distribution Services license, you can use a report to dynamically specify the recipients for a subscription.

To create a dynamic recipient list, you first create a special source report that contains all the necessary information about the recipients of the subscription. You then use the source report to define the dynamic list in MicroStrategy Web. The new dynamic recipient list appears in the list of Available Recipients when defining a new subscription to a standard report or document. When the subscription is executed, only the addresses returned by the source report are included in the delivery.

© 2016, MicroStrategy Inc.

The information in the source report includes email addresses, user IDs, and chosen devices to which to deliver standard MicroStrategy reports and documents. Each address in the source report must be linked to a MicroStrategy user. Any security filters and access control lists (ACLs) that are applied to the address's linked user are also applied to any reports and documents that are sent to the address.

If you have existing Narrowcast Server subscriptions, this feature contains an option in the Select Reports dialog box that allows you to use Narrowcast Server source reports. Narrowcast Server source reports contained subscription information in the page-by elements. When you create a source report to support a dynamic recipient list, you can designate the page-by elements as the location where the system should locate subscription information, thus enabling you to reuse your existing Narrowcast Server source reports. Steps to choose this option when creating a dynamic recipient list are in the *MicroStrategy Web Help*.

The procedure below describes how to create a source report that provides the physical addresses, linked MicroStrategy user IDs, and device type information necessary to create a dynamic recipient list. For steps to create a dynamic recipient list using this source report, see the *MicroStrategy Web Help*.

Prerequisites

- You must have a Distribution Services license.
- To create a dynamic recipient list, you must have the Create Dynamic Address List privilege.
- To subscribe a dynamic address list to a report or document, you must have the Subscribe Dynamic Address List privilege.

To create a source report to support a dynamic recipient list

1. In MicroStrategy Web, create a grid report containing at least three columns. The columns correspond with each of the three required subscription properties:
   - Physical address. For example, this might be provided by a customer email attribute form of the Customer attribute.
   - A MicroStrategy user ID to be linked to the address. For example, this might be provided by a customer ID attribute form of the Customer attribute.
   - Device. This attribute form uses a 32-character hexadecimal string. For example, this may be provided by a preferred format/device attribute form of the Customer attribute.

The data type for the user ID and device columns must be VARCHAR, not CHAR.

 © 2016, MicroStrategy Inc.

2 Save the report with a name and description that makes the report's purpose as a source report for a dynamic recipient list clear.

3 You can now use this source report to create a new dynamic recipient list in MicroStrategy Web. For steps to create a dynamic recipient list using this source report, see the *MicroStrategy Web Help*.

Personalizing email and file subscriptions

You can personalize your email and file subscriptions with macros in the File Name, Subject, Message, ZIP File Name, and Subfolder fields. These macros are automatically replaced with the appropriate text when the subscription is delivered.

For example, you create an email subscription to a report named Daily Revenue. You want the subject of the email to include the name of the report. In the Subscription Editor, in the **Subject** field, you type `Subscription To {&ContentName}`. When the report is delivered, the subject of the email is Subscription to Daily Revenue. Later, the report is changed to include profit, and the name of the report is changed to Daily Revenue and Profit. The subscription is now delivered with the subject Subscription to Daily Revenue and Profit, without any change to the subscription.

You can also use macros to personalize the delivery location and backup delivery location for a file device. For details, including a list of the macros available for file devices, see *Personalizing file locations, page 515*.

The table below lists the macros that can be used in email and file subscriptions, and the fields in which they can be used:

Description	Macro	Fields
Date the subscription is sent	`{&Date}`	Subject, File Name
Time the subscription is sent	`{&Time}`	Subject, File Name
Name of the recipient	`{&RecipientName}`	Subject, File Name
User login	`{&UserLogin}`	All fields
Name of the subscription	`{&Subscription}`	All fields
Project that contains the subscribed report/document	`{&Project}`	All fields
Name of a prompt in the subscribed report/document	`{&Prompt`*`Number`*`&}` (where *`Number`* is the number of the prompt)	All fields
Name of the subscribed report/document	`{&ContentName}`	All fields
Report or document details for the subscribed report/document	`{&ContentDetails}`	Subject, Message

Description	Macro	Fields
Name of the attribute used for bursting (file subscriptions)	`{[Attribute Name]@[Attribute Form]}`	File Name
Name of the attribute used for creating subfolders when bursting (file subscriptions)	`{[Attribute Name]@[Attribute Form]}`	Sub-folder (bursting)

Delivering parts of reports across multiple files: Bursting file subscriptions

Large MicroStrategy reports and documents are often broken up into separate pages by attributes. In a similar way, with Distribution Services, you can split up, or burst, a report or document into multiple files. When the subscription is executed, a separate file is created for each element of each attribute selected for bursting. Each file has a portion of data according to the attributes used to group data in the report (page-by axis) or document (group-by axis).

For example, you may have a report with information for all regions. You could place Region in the page-by axis and burst the file subscription into the separate regions. This creates one report file for each region.

As a second example, if you choose to burst your report using the Region and Category attributes, a separate file is created for each combination of Region and Category, such as Central and Books as a report, Central and Electronics as another, and so on.

When creating the subscription for PDF, Excel, plain text, and CSV file formats, you can use macros to ensure that each file has a unique name. For example, if you choose to burst your document using the Region and Category attributes, you can provide `{[Region]@[DESC]}, {[Category]@[DESC]}` as the file name. When the subscription is executed, each file name begins with the names of the attribute elements used to generate the file, such as Central, Books or Central, Electronics.

To burst a file subscription across multiple files

1. Create a file subscription in MicroStrategy Web by following the steps in *To create a subscription in MicroStrategy Web, page 492*, or edit an existing file subscription in MicroStrategy Web.
2. In the Subscription Editor, click **Burst**... The Select Bursting Criteria options are displayed. All attributes used to group data in the report or document are shown in the **Available Attributes** list.
3. From the **Available Attributes** list, select the attributes to use to break up the data, then click the right arrow to move those attributes to the **Selected Attributes** list.
4. To change the order of attributes for bursting, select an attribute in the **Selected Attributes** list, then click the up or down arrow.

 © 2016, MicroStrategy Inc.

5 In the **File Name** field, type a name for the burst files. You can use macros to ensure that each file has a unique name.

6 Click **OK** to save the subscription.

Delivering parts of reports across multiple files: Bursting file subscriptions to subfolders

Large MicroStrategy reports and documents can be divided into separate pages by attributes. In a similar way with Distribution Services, you can break, or burst, a report or document into multiple subfolders, with each subfolder containing report or document with a portion of data divided by the attributes in the report's page-by or the document's group-by axis. When the subscription is executed, subfolders are dynamically created, if they do not already exist, with the names of the attribute elements. To do this, you provide macro text as part of the bursting subfolder name when creating the file subscription. Each attribute in the macro uses the syntax `{[Attribute Name]@[Attribute Form]}`.

For example, if your report has Manager in the page-by axis, you may burst the report into subfolders using the Manager's last name. In this case, you provide macro text `{[Manager]@[Last Name]}` as the bursting subfolder name.

You can create multiple levels of subfolders if your report or document is grouped by multiple attributes. As a second example, you could have Manager folders with Category subfolders in each. This macro text may be entered in the subfolder name as `{[Manager]@[Last Name]}-{[Manager]@[First Name]}\{[Category]@[DESC]}`. The result of this bursting example is shown in the image below. One of the subscribed reports with books data is in the Books subfolder in the manager's subfolder named Abram-Crisby.

In the example above, the Reports\FileDev1 path was defined as part of the file device used for the subscription. The file name has the date and time appended to the report name because the file device definition has the **Append timestamp to file name** check box selected.

To burst a file subscription across multiple subfolders

1 Create a file subscription in MicroStrategy Web by following the steps in *To create a subscription in MicroStrategy Web, page 492* or edit an existing file subscription in MicroStrategy Web.

2 In the Subscription Editor, click **Burst**... The Select Bursting Criteria options are displayed. All attributes used to group data in the report or document are shown in the **Available Attributes** list.

3 From the **Available Attributes** list, select any attribute to use to create the subfolders, then click the right arrow to move the attribute to the **Selected Attributes** list. The Sub-folder field displays below or to the right of the File Name field.

4 To change the order of attributes for bursting, select an attribute in the **Selected Attributes** list, then click the up or down arrow.

© 2016, MicroStrategy Inc.

5 In the **File Name** field, type a name for the files to be created. You can use macros to ensure that each file has a unique name.

6 In the **Sub-folder** field (the one below or to the right of the File Name field), type a macro to dynamically create the subfolders.

7 Click **OK** to save the subscription.

Managing subscriptions

This section contains the following information:

- *Tasks for managing subscriptions, page 498*
- *Administering subscriptions, page 499*
- *Result caches and subscriptions, page 500*
- *Subscriptions in a clustered system, page 500*

Tasks for managing subscriptions

The table below lists common subscription and delivery-related tasks that users or administrators can perform, and where to perform those tasks, in both MicroStrategy Web and Developer. Note that some tasks can be performed only in MicroStrategy Web.

The steps in the table take you to the main interface to complete the task. For detailed steps, click **Help** once you are in the main interface.

User task	Where to perform the task
Define preferences for a report/document to be delivered to a user's History List folder, mobile device, or system cache.	• In MicroStrategy Web: In a report or document, from the **Report Home** or **Document Home** menu, select **Subscribe to**, then select **History List** or **Mobile**. • In Developer: In a report or document, from the **File** menu, select **Schedule delivery to**, then select **History List**, **Update cache**, or **Mobile**.
Define preferences for a report/document to be delivered to an email address, network storage location, FTP location, or printer (Distribution Services only).	• In MicroStrategy Web, from a report or document, from the **Report Home** or **Document Home** menu, select **Subscribe to**, then select **Email**, **File**, **Printer**, or **FTP**.
Define personal subscription preferences to all reports or documents, in one location.	• In MicroStrategy Web: Click **Preferences** on the left of any page. For History List delivery, select **Project Defaults** on the left, then select **History List**. For Email, File, Printer, FTP delivery, select **User Preferences** on the left, then select **Email Addresses**, **File Locations**, **Printer Location**, or **FTP Locations**. • In Developer: From the **Tools** menu, select **My Subscriptions**.

 © 2016, MicroStrategy Inc.

User task	Where to perform the task
Set up alert-based subscriptions.	• In MicroStrategy Web, add an alert to a report; to do this, run a report, right-click a metric on the report, and select **Alerts**. In the Alerts Editor, after you set up the alert, set up the subscription by selecting **Delivery Settings**.
Schedule a report/document to be sent to another user.	• In MicroStrategy Web, if you own Distribution Services: In a report or document, from the **Report Home** or **Document Home** menu, select **Subscribe to**, then select **History List, Mobile**, **Email**, **File**, **Printer**, or **FTP**. • In MicroStrategy Web, if you do not own Distribution Services: In a report or document, from the **Report Home** or **Document Home** menu, select **Add to History List** or **Add to Mobile**. • In Developer: In a report or document, from the **File** menu, select **Schedule delivery to**, then select **History List**, **Update cache**, or **Mobile**.
Unsubscribe from a report or document.	• In MicroStrategy Web: click **My Subscriptions** on the left of any page. In the Unsubscribe column on the right, select a check box and click **Unsubscribe**. • In Developer: From the **Tools** menu, select **My Subscriptions**. Right-click a subscription, then select **Unsubscribe**.
Change subscription details for a report or document.	• In MicroStrategy Web, click **My Subscriptions** on the left of any page. In the Action column, click the **Edit** icon for the report/document whose subscription you want to edit. • In Developer: From the **Tools** menu, select **My Subscriptions**. Right-click a subscription, then select **Edit**.
Configure who can subscribe to each report.	• In MicroStrategy Web, from a report, from the **Tools** menu, select **Report Options**, then select the **Delivery** tab. Choose to allow all users, specified users, or no users to subscribe to the report. For steps, see the help.
Configure who can subscribe to each document.	• In MicroStrategy Web, from a document in Design mode, from the **Tools** menu, select **Document Options**. The Properties dialog box opens. On the left under Document Properties, click **Delivery** and then choose to allow all users, specified users, or no users to subscribe to the document. For steps, see the help.

Administering subscriptions

You can create, remove, or modify subscriptions through the Subscription Manager.

You can set the maximum number of subscriptions of each type that each user can have for each project. This can prevent excessive load on the system when subscriptions are executed. By default, there is no limit to the number of subscriptions. You set these limits in the Project Configuration Editor, in the Governing Rules: Default: Subscriptions category.

To manage your subscriptions in the Subscription Manager

1 In Developer, log in to a project source.

2 Expand **Administration**, then expand **Configuration Managers**, and then select **Subscriptions**. The list of subscriptions for the project source is displayed on the right side.

3 To manage your subscriptions, select from the tasks below:

- To create a subscription, right-click in the Subscription Manager and select **Subscription Creation Wizard**. The Subscription Wizard opens. Follow the instructions in *To create multiple subscriptions at one time in Developer, page 492*.
- To modify a subscription, right-click the subscription and select **Edit**. The Subscription Editor opens with the information for that subscription. Make any changes and click **OK**. For information about each option in the Subscription Editor, click **Help**.
- To delete a subscription, right-click the subscription and select **Delete**. The subscription is deleted.
- To filter the subscriptions that are listed, right-click in the Subscription Manager and select **Filter**. The Subscription Manager Filter dialog box opens. Specify the filtering criteria and click **OK**. For details on each option in the dialog box, click **Help**.

Result caches and subscriptions

By default, if a cache is present for a subscribed report or document, the report or document uses the cache instead of re-executing the report or document. If no cache is present, one is created when the report or document is executed. For information about how result (report or document) caches are used in MicroStrategy, see *Result caches, page 400*.

When you create a subscription, you can force the report or document to re-execute against the warehouse even if a cache is present, by selecting the **Re-run against the warehouse** check box in the Subscription Wizard. You can also prevent the subscription from creating a new cache by selecting the **Do not create or update matching caches** check box.

You can change the default values for these check boxes in the Project Configuration Editor, in the **Caching: Subscription Execution** category.

Subscriptions in a clustered system

When you set up several Intelligence Server machines in a cluster, you can distribute projects across those clustered machines (or nodes) in any configuration. Each node can host a different subset of projects. For more information, including instructions, on clustering Intelligence Servers, see *Chapter 9, Clustering Multiple MicroStrategy Servers*.

Subscriptions in a clustered system are load-balanced across all nodes of the cluster that host the project containing the subscribed report or document. Subscriptions are load-balanced by the number of subscription jobs created. One subscription job is created for each user or user group in the subscription. For example, if User A and User Group B are subscribed to a dashboard, the subscription creates one job for User A, and a second job

 © 2016, MicroStrategy Inc.

for User Group B. In a two-node cluster, the subscription for User A would execute on one node, and the subscription for User Group B would execute on the other node.

Configuring and administering Distribution Services

MicroStrategy Distribution Services provides high-volume and high-efficiency distribution of reports, documents, and dashboards to email addresses, file servers, networked printers, FTP locations, and devices such as mobile phones. Distribution Services also supports various MicroStrategy Mobile-related features.

Distribution Services functionality is set up and enabled by an administrator in Developer and is used by all types of users through subscribing to deliveries in MicroStrategy Web. Administrators can also subscribe one or more users to a delivery.

This section explains the Distribution Services functionality and steps to set it up in your MicroStrategy system. For details to perform specific tasks using Distribution Services functionality, see the appropriate Help for either MicroStrategy Web or Developer. For example, for steps that a MicroStrategy Web user can take to subscribe to a report delivery when certain threshold conditions are met, which will send a report in the form of an alert to the user, see the MicroStrategy Web Help on creating alerts for threshold conditions.

For details about statistics logging for email, file, print, and FTP deliveries, see *Statistics on subscriptions and deliveries, page 779*.

This section contains the following information:

- *Overview of Distribution Services, page 501*
- *Best practices for using Distribution Services, page 504*
- *High-level checklist to set up a report delivery system, page 505*
- *Creating and managing transmitters, page 506*
- *Creating and managing devices, page 512*
- *Creating and managing contacts, page 522*

Overview of Distribution Services

The administrative tools available with Distribution Services let you set up how reports and documents are transformed into emails and files and sent to email recipients, file locations, printers, mobile devices, or FTP servers. These administrative tools include transmitters, devices, and contacts.

© 2016, MicroStrategy Inc.

About the delivery process

The following high-level steps describe how the subscription and delivery process work:

1. In Developer, the administrator creates a transmitter, such as an email transmitter or file transmitter. For example, the email transmitter is set up with the options that you want the system to use at run time (when the email is sent), to reflect your email server and other environment requirements.

   For information about transmitters, see *About transmitters, page 502*.

2. In Developer, the administrator creates a device for each of the transmitters created above. All devices can have specific settings provided to support your environment and user needs; for example, how to handle a delivery if a printer is down for maintenance, or to add the current date to the name of a file that is delivered to a network location regularly.

   For information about devices, see *About devices, page 503*.

3. In Developer, the administrator creates contacts using the transmitters and devices created above. The contacts contain delivery addresses (email addresses, file paths, printer paths, or FTP servers) and are linked to a MicroStrategy user account for security purposes.

   For information about contacts, see *About contacts, page 503*.

4. In MicroStrategy Web, a user identifies a report or document that she would like to subscribe to have delivered to her regularly.

5. In MicroStrategy Web, the user selects her own address from the To drop-down menu. If she chooses, she can select additional addresses for herself, other MicroStrategy users, or other contacts to also receive this report or document using the subscription.

6. In MicroStrategy Web, the user chooses delivery options such as the delivery schedule, a name for the subscription, the delivery format, an expiration date for the subscription, and so on.

7. The subscription is created. The subscribed users and contact addresses receive the subscribed report or document on the schedule that was defined above.

About transmitters

Transmitters are software components that transform MicroStrategy reports and documents into emails and files and send those emails and files to email recipients, network file locations, network printers, or FTP servers. Distribution Services comes with default transmitters that you can use to create devices.

For details on how to create or modify transmitters, see *Creating and managing transmitters, page 506*.

 © 2016, MicroStrategy Inc.

About devices

Devices are instances of transmitters that have specific transmission and delivery properties. Devices specify the delivery format of a MicroStrategy report or document and the transmission process that sends the report or document to users who subscribe to it. For example, if you want to send a report to a file location, you can create a file device (which is based on a File transmitter) that specifies a file delivery location and file properties, such as appending the current date to the file delivery location, timeout settings, and so on.

Distribution Services comes with default email, file, print, and FTP devices that are already set up. You can use the default devices as is, modify their settings according to your requirements, or create your own devices from scratch if you require additional devices with different combinations of properties. For example, you may require one email device to send emails to Microsoft Outlook and a separate device to send emails to web-based email accounts such as Yahoo, Gmail, Hotmail, and so on.

Devices can be created in a direct connection (two-tier) environment, but print and file locations for those devices are not validated by the system. Print and file locations for devices created in server connections (three-tier) are automatically validated by MicroStrategy.

For details on how to create or manage devices, see *Creating and managing devices, page 512*.

About contacts

Contacts provide a user with a set of associated email addresses, file delivery locations, network printer delivery locations, and FTP servers. To make it easier to manage all the addresses and delivery locations for a user, you can create a contact for each address and delivery location. Thus, one user can have multiple contacts associated with his user account. This allows you to, for example, send a long and complex report to a user's work-related email address, and send a shorter, high-level summary report to the same user's home email address so he can quickly check on a status over the weekend. The two email addresses are set up as separate contacts. The complex report is subscribed to by the user's work contact, and the summary report is subscribed to by the user's home contact.

A contact can also be used to schedule report and document delivery to people who are not MicroStrategy users and do not have user accounts. However, the MicroStrategy user that the contact is associated with must have a Distribution Services license. For example, company shareholders may need to receive a financial document every month. Multiple contacts are created, one for each recipient's email address, and then a MicroStrategy user is created—named Shareholder Finances, for example—and all the contacts are linked to that user. The user is subscribed to the document to be delivered on a schedule, and that user's security profile is applied to the document execution and delivery. The addresses that the document is delivered to are the individual shareholders' email addresses that make up each contact linked to the user.

For details on how to create or manage contacts, see *Creating and managing contacts, page 522*.

© 2016, MicroStrategy Inc.

Security and contacts

Contacts allow you to group multiple addresses by linking those contacts to a MicroStrategy user. The user linked to the contacts can have reports and documents subscribed to the contacts, and thus the reports and documents are delivered to selected addresses and delivery locations defined for those contacts. Each contact can be linked to one MicroStrategy user account, so no other MicroStrategy users can see the address used by a contact.

Each contact must be linked to a MicroStrategy user. The security filters and access control lists (ACLs) that are applied to that user are also applied to reports and documents sent to any contacts associated with that user. For example, if a user has a security filter that restricts her from viewing sales data about regions other than her own, subscriptions delivered to any contacts associated with her do not contain sales data about regions other than her region.

When you create a source report for a dynamic recipient list, you must specify a linked MicroStrategy user for each address. Any security filters and ACLs that are applied to the address's linked user are also applied to any reports and documents sent to the address. For more information about creating a source report for a dynamic recipient list, see *To create a source report to support a dynamic recipient list, page 494*.

Best practices for using Distribution Services

MicroStrategy recommends the following best practices when scheduling Distribution Services subscriptions, in addition to the best practices given above:

- For best results, follow the steps listed in *High-level checklist to set up a report delivery system, page 505*.
- PDF, plain text, and CSV file formats generally offer the fastest delivery performance. Performance can vary, depending on items including your hardware, operating system, network connectivity, and so on.
- The performance of the print delivery method depends on the speed of the printer.
- When sending very large reports or documents:
  - Enable the zipping feature for the subscription so that files are smaller.
  - Use bulk export instead of the CSV file format. Details on bulk exporting are in the *Reports* chapter of the *Advanced Reporting Guide*.
  - Schedule subscription deliveries to occur when your Intelligence Server is experiencing low traffic.
- If your organization is processing a smaller number of subscriptions, such as 100 or fewer, better performance may be achieved by sending each subscription to the largest number of recipients possible. This can be achieved by designing reports or documents that answer business questions for the widest variety of analysts and by adding prompts to the report or document. For an introduction to creating and adding prompts to a report, see the *Basic Reporting Guide*.

 © 2016, MicroStrategy Inc.

For information about how prompts are answered in subscribed reports, see *Prompted reports and subscriptions, page 493*.

If your organization is processing many subscriptions, such as 1,000 or more, better performance may be achieved by sending the largest number of subscriptions possible to the fewest recipients. For example, it may be possible to send all of a team's subscriptions to a project manager, who can then present and distribute the subscribed-to reports in team meetings.

If you are processing many subscriptions, consider using the bulk export feature. Details on bulk exporting are in the *Reports* chapter of the *Advanced Reporting Guide*.

- When creating contacts, make sure that each contact has at least one address for each delivery type. Otherwise the contact does not appear in the list of contacts for subscriptions that are for a delivery type that the contact has no address for. For example, if a contact does not have an email address, when an email subscription is being created, that contact does not appear in the list of contacts.
- When selecting reports to be subscribed to, make sure none of the reports have prompts that require an answer and have no default answer. If a report has a prompt that requires an answer but has no default answer, the subscription cannot run the report successfully, and the subscription is automatically removed from the system.
- Use macros to dynamically specify the delivery location and backup delivery location for a file device (see *Personalizing file locations, page 515*).

High-level checklist to set up a report delivery system

The following high-level checklist describes what you need to do to set up a report delivery system in MicroStrategy using Distribution Services.

Prerequisites

- Understand your users' requirements for subscribing to reports and where they want them delivered.
- Have administrator privileges.
- Have a license to use Distribution Services.
- If you use MicroStrategy Narrowcast Services, during your upgrade to MicroStrategy Distribution Services be sure to use the Migrate Subscriptions for Web Deliveries wizard. This wizard is available from the **MicroStrategy** Developer **Tools** menu. For details on each option in the wizard, click **Help**.

For complete steps to perform a MicroStrategy upgrade, see the *MicroStrategy Upgrade Guide*.

1. Modify existing transmitters or create new transmitters according to your requirements. Distribution Services comes with default email, file, print, mobile, and

FTP transmitters, but if you use these you should modify their settings to suit your environment.

- For best practices for working with transmitters, see *Best practices for working with transmitters, page 507*.
- For steps to modify a transmitter, see *Viewing and modifying a transmitter and accessing the Transmitter Editor, page 507*.
- For steps to create a new transmitter, see *Creating a transmitter, page 508*.

2. Modify existing devices or create new devices according to your requirements. Distribution Services comes with default devices, but if you use these you should modify their settings to suit the systems in your environment.
   - For best practices for working with devices, see *Best practices for working with devices, page 514*.
   - For steps to modify a device, see *Viewing and modifying a device and accessing the device editors, page 514*.
   - For steps to create a new device, see *Creating and managing devices, page 512*.
3. Create contacts so users can subscribe to reports and documents.
   - For best practices for working with contacts, see *Best practices for working with contacts, page 523*.
   - For steps to modify a contact, see *Viewing and modifying a contact and accessing the Contact List area, page 524*.
   - For steps to create a contact, see *Creating a contact, page 525*.

Creating and managing transmitters

A transmitter is a MicroStrategy software component that Distribution Services uses to package subscribed reports and documents into files or emails, and send those files or emails to recipients.

Distribution Services comes with multiple types of transmitters: email (SMTP), file, print, FTP, and mobile. For example, a file transmitter packages and delivers reports in the form of files (PDF, HTML, MS Excel, plain text, and CSV formats) to file storage locations on network computers. A print transmitter sends reports to network printers for printing.

When a user subscribes to a MicroStrategy report, the report is sent to the appropriate transmitter for packaging and delivery. For example, if the report is to be delivered to a file location on a network computer, the report is sent to a file transmitter for conversion to the appropriate file format for delivery. Similarly, if the report is to be delivered in the form of an email to a user's email address, the report is sent to an email transmitter for appropriate packaging and delivery.

A transmitter uses the settings specified in devices to determine how reports are packaged and delivered to the required delivery location. For example, some devices may indicate that reports should be packaged using MIME encoding, but others might specify

 © 2016, MicroStrategy Inc.

UUEncoding. For information on devices and their settings, see *Creating and managing devices, page 512*.

Notification for transmission failures can be configured for email transmitters as described below. Notification for file and print transmission failures can be configured at the project level, using the Project Configuration Editor.

You create and configure transmitters using the Transmitter Editor.

Recommended maintenance tasks for transmitters

- Periodically verify all email addresses where delivery success or failure notification emails are being sent. You can see these email addresses in the Transmitter Editor, on the Notification tab.

Best practices for working with transmitters

- Configure a device to use each type of transmitter and test a delivery using the devices to make sure the transmitters are effective and the devices are working.
- You can easily test an email transmitter by using the **Save to File** check box on the Email Transmitter Editor's Message Output tab.
- To quickly create a new transmitter, duplicate an existing transmitter (such as an out-of-the-box transmitter provided by MicroStrategy), and then change its settings as required.

Viewing and modifying a transmitter and accessing the Transmitter Editor

Using the Transmitter Editor, you can view and modify the definition of a transmitter, rename the transmitter, duplicate the transmitter, and so on.

To view a transmitter or modify its settings

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Transmitters**.
2. In the Transmitter List area on the right, right-click the transmitter that you want to view or change settings for.
3. Select **Edit**. The Transmitter Editor opens.
4. Change the transmitter settings as desired. For details on each option in the interface, click **Help**.
5. Click **OK** to save any changes.

© 2016, MicroStrategy Inc.

Creating a transmitter

In Developer, you can create the following types of transmitters:

- Email: An email transmitter transforms a subscribed report or document and attaches it to an email and sends the email to the inbox of the recipient.
- File: A file transmitter transforms a subscribed report or document into a file (PDF, HTML, Microsoft Excel, plain text, or CSV format) and sends the file to a file storage location such as a folder on a network computer.
- Print: A print transmitter sends the subscribed report or document to a network printer.
- FTP: An FTP transmitter sends the subscribed report or document to an FTP server.
- Mobile: An iPad or iPhone transmitter sends the subscribed report or document to a user's iPad or iPhone.

When a user subscribes to a MicroStrategy report, the report is sent to the appropriate transmitter for packaging and delivery. For example, if the report is to be delivered to a file location on a computer, the report is sent to a file transmitter for packaging and delivery. Similarly, if the report is to be delivered in the form of an email to an email recipient, the report is sent to an email transmitter for packaging and delivery.

You create new transmitters when you need a specific combination of properties and settings for a file, email, print, FTP, or mobile transmitter to package files.

A quick way to create a new transmitter is to duplicate an existing transmitter and then edit its settings to meet the needs for the new transmitter. This is a time-saving method if a similar transmitter already exists, or you want to duplicate the default MicroStrategy transmitter. To duplicate a transmitter, right-click the transmitter that you want to duplicate and select **Duplicate**. For details on each option in the interface, click **Help**.

You create and configure transmitters using the Transmitter Editor.

Creating an email transmitter

An email transmitter creates an email and transforms the subscribed report or document into an attachment to the email, then sends the email to the inbox of the recipients who subscribed to the file.

To create an email transmitter

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Transmitters**.
2. Right-click in the Transmitter List area on the right, select **New**, and select **Transmitter**. The Select Transmitter Type dialog box opens.
3. Select **Email** and click **OK**. The Email Transmitter Editor opens.

 © 2016, MicroStrategy Inc.

4 Change the transmitter settings as desired. For details on each option in the interface, click **Help**.

5 Click **OK** to save the transmitter.

Once an email transmitter is created, you can create email devices that are based on that transmitter. When you create a device, the transmitter appears in the list of existing transmitters in the Select Device Type dialog box. The settings you specified above for the email transmitter apply to all email devices that will be based on the transmitter.

Creating a file transmitter

A file transmitter transforms a subscribed report or document into a file format that the user chooses while subscribing to the report or document. The file transmitter then sends the file to a file storage location on a network computer.

To create a file transmitter

1 From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Transmitters**.

2 Right-click in the Transmitter List area on the right, select **New**, then select **Transmitter**. The Select Transmitter Type dialog box opens.

3 Select **File** and click **OK**. The File Transmitter Editor opens.

4 Change the transmitter settings as desired. For details on each option in the interface, click **Help**.

5 Click **OK** to save the transmitter.

Once a file transmitter is created, you can create file devices that are based on this transmitter. When you create a device, the transmitter appears in the list of existing transmitters in the Select Device Type dialog box. The settings you specified above for the file transmitter apply to all file devices that will be based on the transmitter.

For information on creating a file device, see *Creating a file device, page 515*.

Creating a print transmitter

A print transmitter sends the subscribed report or document to a network printer.

To create a print transmitter

1 From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Transmitters**.

2 Right-click in the Transmitter List area on the right, select **New**, and select **Transmitter**. The Select Transmitter Type dialog box opens.

3 Select **Print** and click **OK**. The Print Transmitter Editor opens.

4 Change the transmitter settings as desired. For details on each option in the interface, click **Help**.

5 Click **OK** to save the transmitter. The new transmitter with the specified name is added to the list of existing transmitters in the Transmitter List area.

Once a print transmitter is created, you can create print devices that are based on the transmitter. When you create a device, the transmitter appears in the list of existing transmitters in the Select Device Type dialog box. The settings you specified above for the print transmitter apply to all print devices that are based on the transmitter.

For information on creating a print device, see *Creating a print device, page 518*.

Creating an FTP transmitter

An FTP transmitter transforms a subscribed report or document into a file format that the user chooses while subscribing to the report or document. The FTP transmitter then sends the file to a location on an FTP server.

To create an FTP transmitter

1 From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Transmitters**.

2 Right-click in the Transmitter List area on the right, select **New**, then select **Transmitter**. The Select Transmitter Type dialog box opens.

3 Select **FTP** and click **OK**. The FTP Transmitter Editor opens.

4 Change the transmitter settings as desired. For details on each option in the interface, click **Help**.

5 Click **OK** to save the transmitter.

Once an FTP transmitter is created, you can create FTP devices that are based on the transmitter. When you create a device, the transmitter appears in the list of existing transmitters in the Select Device Type dialog box. The settings you specified above for the FTP transmitter apply to all FTP devices that will be based on the transmitter.

For information on creating an FTP device, see *Creating an FTP device, page 519*.

Creating an iPad transmitter

An iPad transmitter is used to push real-time updates of reports or documents to a user's iPad. The transmitter transforms the subscribed report or document into a form that can be displayed on the iPad, then it sends the report or document to the subscriber's iPad.

 © 2016, MicroStrategy Inc.

To create a iPad subscription transmitter

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Transmitters**.
2. Right-click in the Transmitter List area on the right, select **New**, and then **Transmitter**. The Select Transmitter Type dialog box opens.
3. Select **iPad Push Notifications** and click **OK**. The Transmitter Editor: iPad Push Notifications dialog box opens.
4. Specify a name and description for the transmitter. The description should include information about settings for this transmitter to help users distinguish it from other transmitters, so they know when to choose this transmitter when associating devices with it.
5. Click **OK** to save the transmitter.

After an iPad subscription transmitter is created, you can create iPad delivery devices that are based on the transmitter. When you create a device, the transmitter appears in the list of existing transmitters in the Select Device Type dialog box.

For information on creating an iPad device, see *Creating an iPad device, page 519*.

Creating an iPhone transmitter

An iPhone transmitter is used to push real-time updates of reports or documents to a user's iPhone. The transmitter transforms the subscribed report or document into a form that can be displayed on the iPhone, then sends the report or document to the subscriber's iPhone.

To create a iPad subscription transmitter

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Transmitters**.
2. Right-click in the Transmitter List area on the right, select **New**, and then **Transmitter**. The Select Transmitter Type dialog box opens.
3. Select **iPhone Push Notifications** and click **OK**. The Transmitter Editor: iPhone Push Notifications dialog box opens.
4. Specify a name and description for the transmitter. The description should include information about settings for this transmitter to help users distinguish it from other transmitters, so they know when to choose this transmitter when associating devices with it.
5. Click **OK** to save the transmitter.

© 2016, MicroStrategy Inc.

After an iPhone transmitter is created, you can create iPhone delivery devices that are based on the transmitter. When you create a device, the transmitter appears in the list of existing transmitters in the Select Device Type dialog box.

For information on creating an iPhone device, see *Creating an iPhone device, page 520*.

Deleting a transmitter

You can delete a transmitter if you no longer need to use it.

Prerequisite

- You cannot delete a transmitter if devices depend on the transmitter. You must first delete any devices that depend on the transmitter.

To delete a transmitter

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Transmitters**.
2. In the Transmitter List area on the right, right-click the transmitter that you want to delete.
3. Select **Delete**. The Confirm Delete Object message is displayed. See the prerequisite above to be sure you have properly prepared the system to allow the transmitter to be deleted.
4. Click **Yes** to confirm the object deletion.

Creating and managing devices

A device specifies the format of a MicroStrategy report or document and the transmission process to send the report or document to users who subscribe to that report or document.

For example, if you want to send reports via email, and your recipients use an email client such as Microsoft Outlook, you can create a Microsoft Outlook email device that has settings appropriate for working with Outlook. If you need to send reports to a file location on a computer on your network, you can create a file device specifying the network file location. If you want to send reports to a printer on your network, you can create a printer device specifying the network printer location and printer properties.

In Developer, you can create the following types of devices:

- Email: An email device automatically sends a report or document in the form of an email to an email address. It can also send the report in the form of a user-selected file format as an attachment with the email.

 © 2016, MicroStrategy Inc.

- File: A file device automatically sends a MicroStrategy report or document in a file format that the user chooses when subscribing to the report, to a file delivery location on a computer on your network. Users can choose from the following file formats: CSV (comma-separated values), PDF, HTML, MS Excel, and plain text. When a user subscribes to a report or document, the file device sends the report or document to the specified location when the subscription requires it to be sent. You can specify your network file location and file properties for the file device to deliver the file to. For steps to create a file device, see *Creating a file device, page 515*.
- Print: A print device automatically sends a report or document to a specified printer on your network. You can define the printer properties for the default print device or you can use the standard printer defaults. For steps to create a print device, see *Creating a print device, page 518*.
- FTP: An FTP device automatically sends a MicroStrategy report or document, in a file format the user chooses, to a delivery location on an FTP server. Users can choose from the following file formats: CSV (command separated values), PDF, HTML, MS Excel, and plain text. Users subscribe to a report or document, which triggers the FTP device to send the report or document to the specified location when the subscription requires it to be sent. For steps to create an FTP device, see *Creating an FTP device, page 519*.
- Mobile: iPad or iPhone devices automatically send a report or document to a user's iPad or iPhone. For steps to create a mobile device, see *Creating an iPad device, page 519* and *Creating an iPhone device, page 520*. These subscriptions are available if you have MicroStrategy Mobile. For detailed information about mobile subscriptions and MicroStrategy Mobile, see the *MicroStrategy Mobile Design and Administration Guide*.

You create new devices when you need a specific combination of properties and settings for a device to deliver files. You can create a new device in two ways. You can either create a completely new device and enter all the supporting information for the device manually, or you can duplicate an existing device and edit the supporting information so it suits your new device. You create and configure devices using the Device Editor.

Devices can be created in a direct connection (two-tier) mode, but print and file locations for those devices are not validated by the system. Print and file locations for devices created when in server connection mode (three-tier) are automatically validated by MicroStrategy.

Recommended maintenance tasks for devices

- Periodically verify all delivery locations to be sure they are active and available.
  - For file delivery locations, use the Device Editor's File: General tab and File: Advanced Properties tab.
  - For printer locations, use the Device Editor's Print: General tab and Print: Advanced Properties tab.
  - For FTP locations, use the Device Editor's FTP: General tab.

© 2016, MicroStrategy Inc.

- Test a delivery using each device to make sure that the device settings are still effective and any system changes that have occurred do not require changes to any device settings.
- If you experience printing or delivery timeouts, use the Device Editor's File: Advanced Properties tab and Print: Advanced Properties tab to change timeout, retry, and other delivery settings.

Best practices for working with devices

- You can allow users to select their own file delivery or print locations. Use the Device Editor's File: General tab and Print: General tab to allow user-defined file delivery and print locations. Any user-defined location overrides the primary file delivery or print location specified in the **File Location** or **Printer Location** field, which, in turn, overrides any backup file delivery or print location specified in the File: Advanced Properties tab or Print: Advanced Properties tab.
- If you have a new email client that you want to use with Distribution Services functionality, create a new email device and apply settings specific to your new email application. To create a new device quickly, use the **Duplicate** option and then change the device settings so they suit your new email application.
- If you rename a device or change any settings of a device, test the device to make sure that the changes allow the device to deliver reports or documents successfully for users.

Viewing and modifying a device and accessing the device editors

Use the Device Editor to view and modify the definition of a device, rename the device, and so on.

To view a device or change its settings

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Devices**.
2. In the Device List area on the right, right-click the device that you want to view or change settings for, and select **Edit**. The Device Editor opens.
3. Change the device settings as desired. For details on each option in the interface, click **Help**.
4. Click **OK** to save your changes.

To rename a device, right-click the device and select **Rename**. Type a new name, and then press ENTER. When you rename a device, the contacts and subscriptions using the device are updated automatically.

 © 2016, MicroStrategy Inc.

Creating a file device

A file device can automatically send a report or document in the form of a file to a storage location such as a folder on a computer on your network. Users subscribe to a report or document that triggers the file device to send the subscribed report or document to the specified location when the subscription requires it to be sent.

You create a new device when you need a specific combination of properties and settings for a file device to deliver files.

You must specify the file properties and the network file location for the file device to deliver files to. You can include properties for the delivered files such as having the system set the file to Read-only, label it as Archive, and so on.

A quick way to create a new file device is to duplicate an existing device and then edit its settings to meet the needs for this new device. This is a time-saving method if a similar device already exists, or you want to duplicate the default MicroStrategy file device. To duplicate a device, right-click the device that you want to duplicate and select **Duplicate**. For details on each option in the interface, click **Help**

To create a new file device

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Devices**.
2. Right-click in the Device List area on the right, select **New**, and then **Device**. The Select Device Type dialog box opens.
3. Select **File** and click **OK**. The File Device Editor opens.
4. Change the device settings as desired. For details on each option in the interface, click **Help**.
5. Click **OK** to save any changes.

Once the file device is created, it appears in the list of existing file devices when you create an address (in this case, a path to a file storage location such as a folder) for a MicroStrategy user or a contact. You select a file device and assign it to the address you are creating. When a user subscribes to a report to be delivered to this address, the report is delivered to the file delivery location specified in that address, using the delivery settings specified in the associated file device. For details to create an address for a user or to create a contact and add addresses to the contact, click **Help**.

Personalizing file locations

You can dynamically specify the **File Location** and **Backup File Location** in a file device using macros. For example, if you specify the **File Location** as `C:\Reports\{&RecipientName}\`, all subscriptions using that file device are delivered to subfolders of `C:\Reports\`. Subscribed reports or documents for each recipient are delivered to a subfolder with that recipient's name, such as `C:\Reports\Jane Smith\` or `C:\Reports\Hiro Protagonist\`.

© 2016, MicroStrategy Inc.

The table below lists the macros that can be used in the **File Location** and **Backup File Location** fields in a file device:

Description	Macro
Date on which the subscription is sent	`{&Date}`
Time at which the subscription is sent	`{&Time}`
Name of the recipient	`{&RecipientName}`
User ID (32-character GUID) of the recipient	`{&RecipientID}`
Distribution Services address that the subscription is delivered to	`{&AddressName}`
File path that a dynamic recipient list subscription is delivered to	`{&RecipientList Address}`

You can also have a subscription dynamically create subfolders according to attributes in a report's page-by axis or a document's group-by area and place the report or document there. For steps, see *Delivering parts of reports across multiple files: Bursting file subscriptions to subfolders, page 497*.

Delivering files from a UNIX Intelligence Server to a Windows file location

If your Intelligence Server machine is using UNIX or Linux, you can configure your system to deliver files to locations on Windows machines.

This process uses Sharity software to resolve the Windows file location as a mount on the UNIX machine. Intelligence Server can then treat the Windows file location as though it were a UNIX file location.

Prerequisites

- You must have a license for MicroStrategy Distribution Services before you can use file subscriptions.
- Sharity must be installed on the Intelligence Server machine. For information about Sharity, see the Sharity website at: `http://www.obdev.at/products/sharity/index.html`.

To set up file delivery from a UNIX Intelligence Server to a Windows location

1 Make sure Sharity is configured on the Intelligence Server machine.

 © 2016, MicroStrategy Inc.

2 Create a new file device or edit an existing file device (see *Creating a file device, page 515*).

3 In the File Device Editor, on the **Cross-Platform Delivery with Sharity™** tab, select the **Enable delivery from Intelligence Server running on UNIX to Windows** check box.

4 In the **User Name** field, type the Windows network login that is used to access the Windows file location for mounting on the Intelligence Server.

5 In the **Password** field, type the password for that user name.

6 In the **Mount Root** field, type the location on the Intelligence Server machine where the mount is stored. Make sure this is a properly formed UNIX path, using forward slashes / to separate directories. For example:
`/bin/Sharity/Mount1`

7 Click **OK**. The File Device Editor closes and the device is saved.

Creating an email device

An email device automatically sends emails, which contain reports or documents that have been subscribed to by users, or for users by other users or administrators. You create a new email device whenever you need a specific combination of properties and settings to deliver files via email. For example, an email sent through Microsoft Outlook requires a device with settings that are different from an email sent through a web-based email account.

You can specify various MIME options for the emails sent by an email device, such as the type of encoding for the emails, the type of attachments the emails, can support, and so on.

A quick way to create a new email device is to duplicate an existing device and then edit its settings to meet the needs for this new device. This is a time-saving method if you have a similar device already created, or you want to make use of the default MicroStrategy email device. To duplicate a device, right-click the device that you want to duplicate and select **Duplicate**. For details on each option in the interface, click **Help**.

Prerequisites

- An understanding of your organization's email server or other email delivery systems.

To create a new email device

1 From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Devices**.

2 Right-click in any open space in the Device List area on the right, select **New**, and then **Device**. The Select Device Type dialog box opens.

3 Select **Email** and click **OK**. The Email Device Editor opens.

© 2016, MicroStrategy Inc.

4 Change the device settings as desired. For details on each option in the interface, click **Help**.

5 Click **OK** to save any changes.

Once an email device is created, it appears in the list of existing email devices when you create an address for a MicroStrategy user or a contact. You select an email device and assign it to the address you are creating. When a user subscribes to a report to be sent to this address, the report is sent to the email recipient specified in that address, using the delivery settings specified in the associated email device. For details on creating an address for a user or on creating a contact and adding addresses to the contact, click **Help**.

Creating a print device

A print device sends a report or document to a specified network printer, where the report or document is automatically printed. You create a new print device whenever you need a specific combination of properties and settings to deliver files to a printer. You can create a new print device, define new printer properties for the default print device that comes with MicroStrategy, or use the default device with its default printer settings.

If you are creating a print device to deliver a report or document to a dynamic recipient list, you can dynamically specify the printer location using the `{&RecipientListAddress}` macro. When the subscription is run, the macro is replaced with the Physical Address specified in its dynamic recipient list. For more information on dynamic recipient lists, see *Using a report to specify the recipients of a subscription, page 493*.

Prerequisite

- The selected printer must be added to the list of printers on the machine on which Intelligence Server is running.

To create a new print device

1 From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Devices**.

2 Right-click in the Device List area on the right, select **New**, and then **Device**. The Select Device Type dialog box opens.

3 Select **Print** and click **OK**. The Print Device Editor opens.

4 Change the device settings as desired. For details on each option in the interface, click **Help**.

5 Click **OK** to save any changes.

Once a print device is created, it appears in the list of existing print devices when you create an address (in this case, a path to the printer) for a MicroStrategy user or a contact. You select a print device and assign it to the address you are creating. When a

 © 2016, MicroStrategy Inc.

user subscribes to a report to be sent to this address, the report is sent to the printer specified in that address, using the delivery settings specified in the associated print device. For details on creating an address for a user or on creating a contact and adding addresses to the contact, click **Help**.

Creating an FTP device

An FTP device automatically sends a report or document in the form of a file to a location on an FTP server. Users subscribe to a report or document, which triggers the FTP device to send the subscribed report or document to the specified location when the subscription requires it to be sent.

You create a new device whenever you need a specific combination of properties and settings for an FTP device to deliver files.

A quick way to create a new FTP device is to duplicate an existing device and then edit its settings to meet the needs for this new device. This is a time-saving method if you have a similar device already created, or you want to make use of the default MicroStrategy FTP device. To duplicate a device, right-click the device that you want to duplicate and select **Duplicate**. For details on each option in the interface, click **Help**.

To create a new FTP device

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Devices**.
2. Right-click in the Device List area on the right, select **New**, and then **Device**. The Select Device Type dialog box opens.
3. Select **FTP** and click **OK**. The FTP Device Editor opens.
4. Change the device settings as desired. For details on each option in the interface, click **Help**.
5. Click **OK** to save the device.

Once the FTP device is created, it appears in the list of existing FTP devices. When you create an address for a MicroStrategy user or a contact, you can select an FTP device and assign it to the address you are creating. When a user subscribes to a report to be delivered to this address, the report is delivered to the delivery location specified in that address, using the delivery settings specified in the associated FTP device. For details on creating an address for a user or on creating a contact and adding addresses to the contact, click **Help**.

For information on creating contacts, see *Creating and managing contacts, page 522*.

Creating an iPad device

An iPad delivery device is used to automatically push real-time updates of reports or documents to a user's iPad, when the report or document is updated. Users subscribe to a

© 2016, MicroStrategy Inc.

report or document, which triggers the iPad device to send the subscribed report or document to the user's iPad.

You create a new device whenever you need a specific combination of properties and settings for an iPad device to deliver reports or documents.

A quick way to create a new iPad device is to duplicate an existing device and then edit its settings to meet the needs for the new device. This is a time-saving method if you have a similar device already created, or you want to duplicate the default MicroStrategy iPad device. To duplicate a device, right-click the device that you want to duplicate and select **Duplicate**.

To create an iPad device

1. In Developer, from the Folder List on the left, expand **Administration**, then **Delivery Managers**, and select **Devices**.
2. Right-click in the Device List area on the right, select **New**, and then **Device**. The Select Device Type dialog box opens.
3. Select **MobileAPNSForIPad** and click **OK**. The Device Editor: Mobile - iPad Push Notifications dialog box opens.
4. Type in or select values for the device settings. For details on each option in the interface, click **Help**.
5. Click **OK** to save the device.

After an iPad delivery device is created, you see it in the list of existing iPad devices when you create an address for a MicroStrategy user or a contact. You select an iPad device and assign it to the address you are creating. When a user subscribes to a report to be delivered to this address, the report is delivered to the iPad specified in that address, using the delivery settings specified in the associated iPad device.

For information on creating contacts, see *Creating and managing contacts, page 522*.

Creating an iPhone device

An iPhone delivery device is used to automatically push real-time updates of reports or documents to a user's iPhone, when the report or document is updated. An iPhone delivery device automatically sends files that contain reports or documents that have been subscribed to by users.

You create a new device whenever you need a specific combination of properties and settings for an iPhone device to deliver reports or documents.

A quick way to create a new iPhone device is to duplicate an existing device and then edit its settings to meet the needs for the new device. This is a time-saving method if you have a similar device already created, or you want to duplicate the default MicroStrategy iPhone device. To duplicate a device, right-click the device that you want to duplicate and select **Duplicate**.

 © 2016, MicroStrategy Inc.

To create an iPhone device

1. In Developer, from the Folder List on the left, expand **Administration**, then **Delivery Managers**, and select **Devices**.
2. Right-click in the Device List area on the right, select **New**, and then **Device**. The Select Device Type dialog box opens.
3. Select **MobileAPNSForIPhone** and click **OK**. The Device Editor: Mobile - iPhone Push Notifications dialog box opens.
4. Type in or select values for the device settings. For details on each option in the interface, click **Help**.
5. Click **OK** to save the device.

After an iPhone delivery device is created, you see it in the list of existing iPhone devices when you create an address for a MicroStrategy user or a contact. You select an iPhone device and assign it to the address you are creating. When a user subscribes to a report to be delivered to this address, the report is delivered to the iPhone specified in that address, using the delivery settings specified in the associated iPhone device.

For information on creating contacts, see *Creating and managing contacts, page 522*.

Deleting a device

You can delete a device if it is no longer needed.

Prerequisites

Update the contacts and subscriptions that are using the device by replacing the device with a different one. To do this, check whether the device you want to delete is used by any existing addresses:

- To find contacts, from the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Contacts.** In **View Options**, select the device name.
- To find subscriptions that are dependent on the device, right-click each contact and select **Search for dependent subscriptions**.

To delete a device

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Devices**.
2. In the Device List area on the right, right-click the device you want to delete.
3. Select **Delete**. The Confirm Delete Object message is displayed. Be sure that you have properly prepared the system to allow the device to be deleted. See the Prerequisites above.

© 2016, MicroStrategy Inc.

4 Click **Yes** to confirm the object deletion.

Creating and managing contacts

A contact is a MicroStrategy object that represents a delivery address (an email address, a file storage location on a network, a network printer path, or a location on an FTP server). MicroStrategy users select a contact when they subscribe themselves or other users to reports or documents. They do so by simply selecting a contact name, rather than having to specify email addresses, file storage locations, printer locations, or FTP server locations at the time of report subscription. Depending on needs, a user may require multiple delivery locations of different types, such as a work and a personal email address, several file storage locations, and one or more printer or FTP server locations. To make it easier to manage all delivery addresses for your users, you create contacts.

Contacts can also be used when you want to deliver reports or documents to people who are not MicroStrategy users. For an example and more details on using contacts this way, see the *MicroStrategy Developer Help*.

As an alternative to contacts, you can deliver reports and documents by subscribing individual addresses and related personalization that are stored in sources other than the metadata. Dynamic recipient lists are based on a special source report you create that contains email addresses, user IDs, and chosen devices to which to deliver standard MicroStrategy reports and documents. For details, see *Using a report to specify the recipients of a subscription, page 493*.

To be available for subscriptions, a newly created contact must be linked to a MicroStrategy user. The security for a contact and its associated delivery comes from this linked user. The user linked to the contact can have reports and documents subscribed to the contact and thus the reports and documents are delivered to the email address or delivery location in that contact.

Contacts can also be combined into groups so that it is easier to select multiple contacts for a subscription. You can assign a security filter to the contact group, or you can choose to use the security filters linked to the users.

Prerequisites

- Understand your users' requirements for file and printer delivery locations, and email addresses, as well as the reports and documents they are likely to subscribe to or be subscribed to. For example, some MicroStrategy documents are Flash dashboards that require Flash to be installed wherever the dashboard is to be delivered. For requirements for Flash dashboards, see the *MicroStrategy Document Creation Guide*.
- Understand security requirements for any contacts you plan to create. The MicroStrategy user who is linked to the contact defines the security applied to the contact address (email address or file or printer network path).
- Have administrator privileges.

 © 2016, MicroStrategy Inc.

Recommended maintenance tasks for contacts

- Verify delivery addresses to be sure they are active and available. The Physical Address column in the Contact List area displays the file storage locations, email addresses, and printer locations where reports and documents are to be delivered. If the Physical Address column is not visible in the Contact List area, right-click to display it.
- For server outages or maintenance:
  - Change the default delivery location of affected contacts to another available location. You set delivery locations on the Contact Editor's Addresses tab.
  - Alternatively, for short-term server outages, disable the contacts that are set up to deliver to those servers, and then enable them again after the servers are up again. You disable and enable contacts on the Contact Editor's General tab.
- Check the list of MicroStrategy users linked to contacts to verify whether a user has been deleted from the system. If a user has been deleted, you may want to delete or disable the associated contacts.
- If the user linked to one or more contacts does not need to receive subscribed reports and documents, delete any associated contacts.
- If the delivery of a user's subscribed reports or documents needs to be stopped for some time, disable any associated contacts.
- Before deleting a contact, search for subscriptions dependent on the contact by right-clicking the contact and selecting **Search for dependent subscriptions**. Delete those subscriptions after you have deleted the contact.
- Periodically check the status of contacts or contact groups, specifically the list of disabled contacts and contact groups, and review that status. You can filter the contacts so that the Contact List area displays only those contacts that are disabled.

Best practices for working with contacts

- Use Tree View in the Contact List area. This view reflects your organization's structure and helps you visualize how to create contact groups according to users who will receive the same subscribed reports. Tree View can be helpful as your list of contacts grows larger because it is easy to collapse and expand groupings of contacts.
- If you have many contacts and contact groups, use the filter to restrict the number of contacts you are viewing when performing contact maintenance tasks. For steps to use the filter, click **Help**.
- To create a new contact using the fastest method, duplicate an existing contact and then change the details as required.
- When creating many new contacts or maintaining a number of contacts by moving delivery locations (addresses) between them, copy and paste the addresses between the contacts.
- Before disabling or deleting a contact, use the **Search for dependent subscriptions** option to review subscriptions that might be affected by your action.

© 2016, MicroStrategy Inc.

Viewing and modifying a contact and accessing the Contact List area

You can view and modify the definition of a contact, rename the contact, duplicate the contact, delete or disable a contact, and so on, using the Contact Editor.

To view a contact or change its settings

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Contacts**.
2. In the Contact List area on the right, right-click the contact that you want to view or change settings for.
3. Select **Edit**. The Contact Editor opens.
4. Change the name, description, or other settings of the contact. For details on each option, click **Help**.
5. Click **OK** to save your changes.

In addition to modifying the definition of a contact, you can perform the following operations on a contact by right-clicking it:

- **Duplicate**: Creates a copy of the contact. See *Creating a contact, page 525*.
- **Rename**: Allows you to rename the selected contact. Right-click the contact and select **Rename**. Type a new name and then press ENTER.
- **Copy**: Copies the selected address.
- **Paste**: Active only when you have copied an address. Adds the copied address to the selected contact.
- **Delete**: Deletes the selected contact. For important warnings and other details, see *Deleting a contact, page 528*.
- **Disable Contact/Enable Contact**: Disables or enables the selected contact. Disabling a contact means the contact will no longer be available for report or document subscription. For example, this option is useful when a printer or server is down for maintenance and the delivery address (path to the printer or file storage location) associated with the contact is not available for a time.
- **Show Contact Group Membership**: Opens the Member of tab of the Contact Editor for the selected contact. This tab displays a list of contact groups that this contact is a member of.
- **Search for dependent subscriptions**: Displays a list of report and document subscriptions that are dependent on this contact. This option is important to use before you delete a contact.

 © 2016, MicroStrategy Inc.

Creating a contact

You create a new contact for each delivery address that reports and documents will be delivered to.

A quick way to create a new contact is to duplicate an existing contact and then edit its settings to meet the needs for this new contact. This is a time-saving method if a similar contact already exists or if you want to make use of the default MicroStrategy contact. To duplicate a contact, right-click the contact that you want to duplicate and select **Duplicate**. For details on each option in the interface, click **Help**.

To create a contact

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Contacts**.
2. Right-click in the Contact List area on the right, select **New**, and then **Contact**. The Contact Editor opens.
3. Change the contact settings as desired. For details on each option in the interface, click **Help**.
4. Click **OK**.

Once a contact is created, it must be linked to a MicroStrategy user to be available for report or document subscriptions. A contact does not have any credentials or security profile associated with it. Because the system needs a security profile to execute the reports and documents subscribed to a contact, the contact must be linked to a MicroStrategy user for the required security profile. This means that when a subscribed report is sent to a contact address, the report is executed using the credentials and the security profile of the linked user.

To link a contact to a user

1. Double-click a user to open the User Editor.
2. Expand **Deliveries**, then select **Contacts**.
3. Click **Add** to open the Link Contacts to User dialog box.
4. Select one or more contacts to link to the user. Press CTRL to select multiple contacts. You can narrow the list of contacts displayed in the pane by doing one of the following:
   - To search for the contact by name, type the contact name in the search field.
   - Choose a group from the drop-down list. The pane is updated to include only contacts belonging to that group.
5. Click the right arrow **>** to move the contacts to the right.

© 2016, MicroStrategy Inc.

6 Click **OK** to link the contacts and return to the User Editor.

For details on each option in the interface, click **Help**.

Creating a contact group

A contact group is a set of contacts that are combined under one name. Contact groups are useful to create when certain reports must be sent to multiple contacts. For example, if four contacts need to receive the same subscribed reports, you can group the contacts into a contact group and subscribe the contact group to the reports, rather than subscribing each contact individually.

Once a contact group is created, it must be linked to a MicroStrategy user to be available for subscriptions. A contact group without a linked user does not have any login credentials or security profile associated with it. Because the system needs a security profile to execute the reports and documents subscribed to a contact group, the contact group must be linked to a MicroStrategy user for the required security profile. This means that when a subscribed report needs to be sent to a contact group, the report is executed using the credentials and the security profile of the linked user.

A contact group is similar to a Microsoft Outlook distribution list; each member contact of a contact group receives the same subscribed reports when the contact group is chosen as the recipient of a subscription. Therefore, if a contact should not receive certain reports, the contact should not be made a part of a contact group.

To create a contact group

1 From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Contacts**.

2 Right-click in the Contact List area on the right, select **New**, and then **Contact Group**. The Contact Group Editor opens.

3 Change the contact settings as desired. For details on each option in the interface, click **Help**.

A contact group must be linked to a user for its contacts to be available for report and document subscription.

4 Click **OK**.

Placing multiple contact groups in a contact group

You can also group multiple contact groups into one contact group. Grouping multiple contact groups into a contact group makes it easy to send out wide-distribution reports that have no security implications, such as an employee birthday list that is sent out at the beginning of every month.

 © 2016, MicroStrategy Inc.

All members (contacts) of each contact group in the top-level contact group receive the same subscribed reports, when the top-level contact group is chosen as the recipient of a subscription.

Maintaining a linked user

A contact must be linked to a MicroStrategy user to be available for subscriptions. A contact does not have any credentials or security profile associated with it. Because the system needs a security profile to execute the reports and documents subscribed to a contact, the contact must be linked to a MicroStrategy user for the required security profile. This means that when a subscribed report is sent to a contact, the report is executed using the login credentials and the security profile of the linked user.

The Contact List area displays a list of users linked to contacts, along with the list of contacts and contact groups. Right-click a user and select from the following options:

- **Edit**: Opens the User Editor for the selected user. For details on each option in the interface, click **Help**.

  Any changes made to the user account in the User Editor will affect the user's account across the MicroStrategy system.

- **Rename**: Allows you to rename the selected user. Right-click the user and select **Rename**. Type a new name and press ENTER.
- **Copy**: Copies the selected delivery location (address).
- **Paste**: Active only when you have copied a delivery location (address). Adds the copied address to the selected user.
- **Delete**: Deletes user's delivery locations (addresses). For important prerequisites and other details, see *Deleting a contact, page 528*.
- **Disable Contact**: Disables user's delivery locations (addresses). If a user's addresses are disabled, the user cannot be subscribed to reports. If some reports are already subscribed to the disabled contacts and their associated addresses, the reports are not delivered to the addresses unless the contact is enabled again.

  Disabling a user's addresses does not disable the user account in MicroStrategy.

- **Show Contact Group Membership**: Displays a list of contact groups to which the user belongs.
- **Search for dependent subscriptions**: Displays a list of reports and documents subscribed to by this user.

Maintaining addresses

An address is any delivery location for a report or document subscription. An address can be the path to a folder on a network or FTP server for file storage, a printer path, or an email address.

© 2016, MicroStrategy Inc.

A MicroStrategy user can have several email, file, printer, or FTP server addresses for subscribed reports to be delivered to when the user subscribes to or is subscribed to a report. Contacts (each containing an address) are linked to the user. You can create and add addresses for a user in the **Deliveries**: **Addresses** category of the User Editor.

In the Contacts List area, right-click an address for a contact or a user and select from the following options:

- **Edit**: Opens the User Editor to the Deliveries: Addresses category if you right-click an address within a user. Opens the Contact Editor: Addresses tab if you right-click an address within a contact. For details on each option in the interface, click **Help**.
- **Rename**: Allows you to rename the selected address. Right-click the address and select **Rename**. Type a new name and press ENTER.
- **Copy**: Copies the selected address.
- **Paste**: Adds the copied address to the contact or the user.
- **Delete**: Deletes the selected address.
- **Search for dependent subscriptions**: Displays a list of reports and documents dependent on this address.

Deleting a contact

You can delete a contact if you no longer want subscribed reports or documents to be delivered to the address within that contact. Depending on your reasons for deleting a contact, an alternative may be to disable the contact. For example, if you want report deliveries to a contact to be stopped for some time, you may disable the contact rather than deleting it. Later, you can easily and quickly enable the contact to resume the report deliveries to that contact.

Prerequisites

- Check to see whether you need to save any of the delivery locations (addresses) that make up the contact that you plan to delete. To do this, first search for subscriptions that are dependent on the contact by right-clicking the contact and selecting **Search for dependent subscriptions**. If you want those subscriptions to continue to be sent to any of the contact's delivery locations, create a new contact and then copy/paste that delivery location into the new contact.

To delete a contact

1. From the Developer Folder List, expand **Administration**, expand **Delivery Managers**, and select **Contacts**.
2. In the Contact List area on the right, right-click the contact you want to delete.
3. Select **Delete**. The Confirm Delete Object message is displayed.
4. Click **Yes** to confirm the object deletion.

 © 2016, MicroStrategy Inc.

13

ADMINISTERING MICROSTRATEGY WEB AND MOBILE

Introduction

As a MicroStrategy system administrator, you may be responsible for managing MicroStrategy Web and Mobile environments. Some of these tasks are performed in the Developer interface, such as managing user and group privileges for Web users, or registering a project in server (3-tier) mode so it can be available in Web. However, other administrative parameters are set using the MicroStrategy Web or Mobile Server administrative interface. In addition, configuring your mobile devices can be done through the Mobile Server Administrator.

In addition to the information in this chapter, each option in the MicroStrategy Web or Mobile Server administration interface is documented in the relevant Help system. To access this Help system, click **Help** in MicroStrategy Web Administrator or Mobile Server Administrator.

Topics covered in this chapter are:

- *Assigning privileges for MicroStrategy Web, page 530*
- *Using the MicroStrategy Web Administrator page, page 531*
- *Defining project defaults, page 533*
- *Using additional security features for MicroStrategy Web and Mobile, page 534*

© 2016, MicroStrategy Inc.

- *Integrating Narrowcast Server with MicroStrategy Web products, page 541*
- *Enabling users to install MicroStrategy Office from Web, page 542*
- *FAQs for configuring and tuning MicroStrategy Web products, page 542*

Assigning privileges for MicroStrategy Web

MicroStrategy Web products are available in three different editions, each having an associated set of privileges. The number of users assignable to any one edition is based on your license agreement.

MicroStrategy provides these editions for MicroStrategy Web products:

- Web Professional or Web Universal Professional
- Web Analyst or Web Universal Analyst
- Web Reporter or Web Universal Reporter

The privileges available in each edition are listed in the *List of Privileges* chapter in the *Supplemental Reference for System Administration*. You can also print a report of all privileges assigned to each user based on license type; to do this, see *Auditing your system for the proper licenses, page 208*.

All MicroStrategy Web users that are licensed for MicroStrategy Report Services may view and interact with a document in Flash Mode. Certain interactions in Flash Mode have additional licensing requirements:

- Users are required to license MicroStrategy Web Analyst to pivot row or column position in a grid or cross-tabular grid of data in Flash Mode.
- Users are required to license MicroStrategy Web Professional to modify the properties of Widgets used in a document in Flash Mode.

A user assigned to an edition is entitled to a complete set, or identified subset, of the privileges listed for that edition.

If a user is assigned to multiple user groups, the privileges of those groups are additive, and determine the edition usage of that particular user. For example, if a user is a member of both the Finance and the Accounting user groups, privileges for that user are equivalent to the cumulative set of privileges assigned to those two groups.

One privilege, Web Administration, can be assigned to any edition of Web user. This privilege allows the user to access the Web Administrator page to manage server connections, and to access the Project defaults link on the Preferences page to set defaults for all users.

The MicroStrategy security model enables you to set up user groups that can have subgroups within them, thus creating a hierarchy. The following applies to the creation of user subgroups:

© 2016, MicroStrategy Inc.

- A child subgroup automatically inherits privileges assigned to its parent group.
- A child subgroup can be assigned other privileges in addition to inherited privileges.

User groups corresponding to the three editions of MicroStrategy Web products are predefined with the appropriate privilege sets. These user groups are available under the User Group folder in the Administration folder for your project.

You need project administration privileges to view and modify user group definitions.

See your license agreement as you determine how each user is assigned to a given privilege set. MicroStrategy Web products provide three Web editions (Professional, Analyst, Reporter), defined by the privilege set assigned to each.

Assigning privileges outside those designated for each edition changes the user's edition. For example, if you assign to a user in a Web Reporter group a privilege available only to a Web Analyst, MicroStrategy considers the user to be a Web Analyst user.

Within any edition, privileges can be removed for specific users or user groups. For more information about security and privileges, see *Chapter 2, Setting Up User Security*.

License Manager enables you to perform a self-audit of your user base and, therefore, helps you understand how your licenses are being used. For more information, see *Auditing and updating licenses, page 206*.

Using the MicroStrategy Web Administrator page

The Administrator page allows you to connect and disconnect MicroStrategy Web or Web Universal products to Intelligence Servers. It also lets you set MicroStrategy Web products governors such as server and request time-outs and the maximum number of connections MicroStrategy Web products can make to an Intelligence Server.

If you have the appropriate privileges, you can find the link to the Administrator page on the MicroStrategy Web or Web Universal home page.

You can also configure MicroStrategy Web products to recognize MicroStrategy Narrowcast Server and enable the **Scheduled Delivery** and **Send Now** options. For more information about enabling these features, see *Integrating Narrowcast Server with MicroStrategy Web products, page 541*.

Controlling access to the Administrator page

You can restrict access to the Web Administrator page in two ways:

- Assign the Web Administration privilege to select users
- Use Microsoft IIS and Windows security to limit access to the page file

For information about controlling access to this page when using different Web servers, see the *MicroStrategy Installation and Configuration Guide*. Specifically, see the section "Launch mstrWebAdmin servlet" or "Controlling access to the Web Administrator page" according to the platform you are using.

Assigning the Web Administration privilege

The link to the Administrator page appears only if at least one of the following criteria is true:

- You are logged in to a project and have the Web Administration privilege.
- Your MicroStrategy Web product is not connected to any Intelligence Servers. In this case, there is no way to tell whether you have the Web Administration privilege because there is no Intelligence Server to verify your credentials. However, once you connect to an Intelligence Server, you do not see the link unless you log in to a project in which you have the Web Administration privilege.

For steps on how to assign this privilege to a user, see *Setting Up User Security, page 51*.

Using Microsoft IIS and Windows security for MicroStrategy Web products (ASP.NET)

Users without the Web Administration privilege cannot access the Administrator page from within MicroStrategy Web. However, this does not prevent someone from simply typing the URL in a Web browser to navigate to the Administrator page. To prevent this from happening with the ASP.NET version of MicroStrategy Web products using Microsoft IIS, you must limit access to the file itself using Microsoft IIS and Windows security. The default location of the Administrator page file is `Program Files (x86) \ MicroStrategy\WebAspx\asp\Admin.aspx`

In the J2EE version, the Administrator page is a servlet and access to the servlet is controlled using the Web and application servers. The default location of the Administrator servlet varies depending on the platform you are using. For details, see the *MicroStrategy Installation and Configuration Guide*.

Using security roles with Web Universal

In MicroStrategy Web Universal, when using the J2EE version, users must have the proper user ID and password to access the Administrator servlet (`mstrWebAdmin`). Consult the documentation for your particular Web and application servers for information about file-level security requirements and security roles.

The default location of the Administrator servlet varies depending on the platform you are using. For details, see the *MicroStrategy Installation and Configuration Guide*.

 © 2016, MicroStrategy Inc.

Defining project defaults

If you have the Web Administration privilege, you can set the default options for one or more projects in the Preferences section. The Project defaults link is displayed only if you have the Web Administration privilege.

Any changes you make to the project defaults become the default settings for the current project or for all Web projects if you select the **Apply to all projects on the current MicroStrategy Intelligence Server (server name)** option from the drop-down list.

The project defaults include user preference options, which each user can override, and other project default settings accessible only to the administrator.

For information on the History List settings, see *Saving report results: History List, page 425*.

Loading and applying default values

The Load Default Values option works differently on the Project defaults and the User preferences pages:

- When the administrator who is setting the Project defaults clicks **Load Default Values**, the original values shipped with the MicroStrategy Web products are loaded on the page.
- When users who are setting User preferences click **Load Default Values**, the project default values that the administrator set on the Project defaults pages are loaded.

The settings are not saved until you click **Apply**. If you select **Apply to all projects on the current Intelligence Server (server name)** from the drop-down menu, the settings are applied to all projects, not just the one you are currently configuring.

Setting user preferences and project defaults

Users can change the individual settings for their user preference options by accessing them via the Preferences link at the top of the Web page. However, you can set what default values the users see for these options. To do this, click the **Preferences** link, then click the **Project defaults** link on the left-hand side of the page (under the “Preferences Level” heading).

You can then set the defaults for several categories, including the following:

- General
- Folder Browsing
- Grid display
- Graph display
- History List

© 2016, MicroStrategy Inc.

- Export
- Print (PDF)
- Drill mode
- Prompts
- Report Services
- Security (see note below for Web Universal)
- Project display (see note below for Web Universal)
- Office

Some of the following categories are displayed only in certain circumstances. For example, the Report Services link appears only if you have a license to use Report Services.

Each category has its own page and includes related settings that are accessible only to users with the Web Administration privilege. For details on each setting, see the MicroStrategy Web Help for the Web Administrator.

Using additional security features for MicroStrategy Web and Mobile

This section describes how MicroStrategy Web and Mobile products can be made more secure by using standard Internet security technologies such as firewalls, digital certificates, and encryption.

For information on enabling secure, encrypted communications for Web, see *Chapter , Enabling Secure Communication*.

Using firewalls

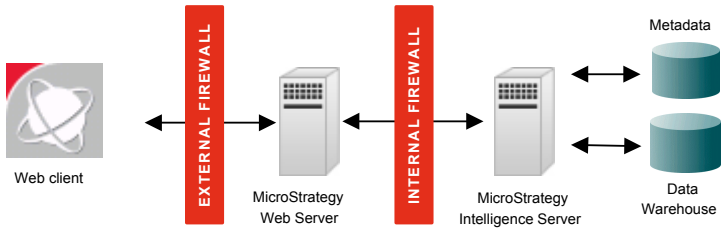
A *firewall* enforces an access control policy between two systems. A firewall can be thought of as something that exists to block certain network traffic while permitting other network traffic. Though the actual means by which this is accomplished varies widely, firewalls can be implemented using both hardware and software, or a combination of both.

Firewalls are most frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. If you use MicroStrategy Web or Mobile products over the Internet to access projects on an Intelligence Server that is most likely on an intranet, there is the possibility that a malicious user can exploit the security hole created by the connection between the two systems.

 © 2016, MicroStrategy Inc.

Therefore, in many environments and for a variety of reasons you may wish to put a firewall between your Web servers and the Intelligence Server or cluster. This does not pose any problems for the MicroStrategy system, but there are some things you need to know to ensure that the system functions as expected.

Another common place for a firewall is between the Web clients and the Web or Mobile server. The following diagram shows how a MicroStrategy system might look with firewalls in both of these locations:



Regardless of how you choose to implement your firewalls, you must make sure that the clients can communicate with MicroStrategy Web and Mobile Servers, that MicroStrategy Web and Mobile can communicate with Intelligence Server, and vice versa. To do this, certain communication ports must be open on the server machines and the firewalls must allow Web server and Intelligence Server communications to go through on those ports. Most firewalls have some way to specify this. Consult the documentation that came with your firewall solution for details.

To enable communication through a firewall

1. Client Web browsers communicate with MicroStrategy Web on port 80 (HTTP). So, if you have a firewall between your clients and MicroStrategy Web servers, you must make sure port 80 is allowed to send and receive requests through the firewall.

   Depending on how you deployed Web Universal, it may communicate on a different port number.

2. MicroStrategy Web products can communicate with Intelligence Server using any port that is greater than 1024. By default, the port used is 34952. So, if you have a firewall between your Web servers and Intelligence Server, you must make sure port 34952 is allowed to send and receive TCP/IP requests through the firewall.

   You can change this port number if you wish. See the steps in the next procedure *To change the port through which MicroStrategy Web and Intelligence Server communicate, page 536* to learn how.

3. You must configure your firewall to allow MicroStrategy Web products to communicate with Intelligence Server using port 3333. This is in addition to the port configured in the previous step of this procedure.

4. The MicroStrategy Listener Service communicates with MicroStrategy Web products and Intelligence Server on port 30172. So, if you are using the Listener Service, you

must make sure port 30172 is allowed to send and receive TCP/IP and UDP requests through the firewall. You cannot change this port number.

To change the port through which MicroStrategy Web and Intelligence Server communicate

1 By default, MicroStrategy Web and Intelligence Server communicate with each other using port 34952 (Web Universal may use a different port depending on how you deployed it). If you wish to change this, you must change it for both the Web servers and the Intelligence Servers. The port numbers on both sides must match.

If you are using clusters, you must make sure that all machines in the Web server cluster can communicate with all machines in the Intelligence Server cluster.

To change the port number for Intelligence Server

2 In Developer, log in to the project source that connects to the server whose port you want to change.

3 In the Service Manager, click **Options**.

4 On the Intelligence Server Options tab, type the port number you wish to use in the **Port Number** box. Save your changes.

5 A message appears telling you to restart Intelligence Server. Click **OK** to continue.

6 Restart Intelligence Server.

7 In Developer, right-click the project source that connects to the Intelligence Server whose port number you changed and choose **Modify Project Source**. The Project Source Manager opens.

8 On the Connection tab, enter the new port number and click **OK** to save your changes.

You must update this port number for all project sources in your system that connect to this Intelligence Server.

To change the port number for MicroStrategy Web

9 Open the Administrator page in MicroStrategy Web.

10 If your MicroStrategy Web product is connected to the Intelligence Server whose port number you changed, click **Disconnect** to disconnect it. You cannot change the port while connected to an Intelligence Server.

It probably is not connected because the MicroStrategy Web product does not yet know the new port number you assigned to Intelligence Server.

 © 2016, MicroStrategy Inc.

11 In the entry that corresponds to the appropriate Intelligence Server, click **Modify** (in the Properties column, all the way to the right). The Server property detail page opens.

12 In the **Port** box, type the port number you wish to use. This port number must match the port number you set for Intelligence Server. An entry of 0 means use port 34952 (the default).

13 Click **Save** to save your changes and you can connect to Intelligence Server again.

If the port numbers for your MicroStrategy Web product and Intelligence Server do not match, you get an error when the MicroStrategy Web product tries to connect to Intelligence Server.

For detailed steps for any of the high-level steps listed above, see the *MicroStrategy Developer Help*.

Using cookies

A *cookie* is a piece of information that is sent to your Web browser—along with an HTML page—when you access a Web site or page. When a cookie arrives, your browser saves this information to a file on your hard drive. When you return to the site or page, some of the stored information is sent back to the Web server, along with your new request. This information is usually used to remember details about what a user did on a particular site or page for the purpose of providing a more personal experience for the user. For example, you have probably visited a site such as Amazon.com and found that the site recognizes you. It may know that you have been there before, when you last visited, and maybe even what you were looking at the last time you visited.

MicroStrategy Web products use cookies for a wide variety of things. In fact, they use them for so many things that the application cannot work without them. Cookies are used to hold information about user sessions, preferences, available projects, language settings, window sizes, and so on. For a complete and detailed reference of all cookies used in MicroStrategy Web and MicroStrategy Web Universal, see the *MicroStrategy Web Cookies* chapter in the *Supplemental Reference for System Administration*. The sections below describe the cookie related settings available in each product.

Using cookies in MicroStrategy Web products

MicroStrategy Web products provide several options related to the use of cookies. All of these settings can be found on the Web Administrator's Security Page.

- Cookies Security Level:
  - **Disable cookies**: The application does not store any cookies. This means that no settings are stored in cookies; instead, the application stores persistable settings (for example, the open and close state of a view filter) in the metadata. To make your application highly secure, you can select this option.
  - **Enable cookies**: The application stores some of your settings using browser cookies. This is the default option. Temporary settings stored in cookies are lost when you close the browser; these settings include the last page visited, filter and

template definitions for report execution, shared report location, and so on. Persistable settings are stored in cookies and can be restored when you visit the application in the future, for example when you open or close editors.

If you enable cookies, you also have the option to enable or disable:

- **Store MicroStrategy Intelligence Server sessions information in temporary cookies instead of on the Web Server**: This option specifies whether Intelligence Server session information should be saved in cookies or not. Since the Intelligence Server session information is sensitive, it is not secure to turn on this option in most cases. This is checked in those cases when a cluster is set up and does not automatically handle session replication. The session replication is the distribution of the session information on the client instead of on the Web server so that the user can connect seamlessly to any of the Intelligence Server machines.

When you enable this, temporary information such as Session ID of the Intelligence Server sessions is saved.

If the Disable cookies option is selected, selecting this option does not save any information in the cookies.

Using encryption

Encryption is the translation of data into a sort of secret code for security purposes. The most common use of encryption is for information that is sent across a network so that a malicious user cannot gain anything from intercepting a network communication. Sometimes information stored in or written to a file is encrypted. The SSL technology described earlier is one example of an encryption technology.

MicroStrategy Web products can use encryption in many places, but by default, most are not used unless you enable them.

Encryption in MicroStrategy Web products

You can encrypt all communication between the Web server and Intelligence Server. Additional overhead is involved in encrypting and decrypting all this network traffic so you may see a noticeable performance degradation if encryption is enabled. However, if you are working with sensitive or confidential information, this may be an acceptable trade-off.

To encrypt all communication between MicroStrategy Web products and Intelligence Server

1. Go to the Administrator Page.
2. At the top of the page or in the column on the left, click **Security** to see the security settings.

 © 2016, MicroStrategy Inc.

3 Within the Encryption area, select one of the following encryption options:

- **No encryption** (default): Data between Web and Intelligence Server is not encrypted.
- **AES Tunneling**: Uses the AES encryption algorithm to secure data between Web and Intelligence Server. This option increases the load on the Intelligence Server and this may result in reduced performance in MicroStrategy Web.
- **SSL**: Uses Secure Socket Layer (SSL) encryption to secure data between Web and Intelligence Server. This is the recommended option for secure communications. If you want Web to verify Intelligence Server's SSL certificate before setting up a connection, select **Validate Intelligence Server certificate**. For instructions to set up SSL encryption for Web, see *Chapter , Enabling Secure Communication*.

4 Click **Save** to save your changes. Now all communication between the Web server and Intelligence Server is encrypted.

Applying file-level security

It is important to remember that no matter what kind of security you set up, there is always the possibility that a malicious user can bypass it all simply by gaining access to the physical machine that hosts the Web application. For this reason you should make sure that the machine is in a secure location and that you restrict access to the files stored on it using the standard file-level security offered by the operating system.

In typical production environments, only a small number of administrative users are allowed to log on to server machines. All other users either have very limited access to the files and applications on the machine or, better yet, no access at all.

For example, with Microsoft IIS, by default only the “Internet guest user” needs access to the virtual directory. This is the account under which all file access occurs for Web applications. In this case, the Internet guest user needs the following privileges to the virtual directory: read, write, read and execute, list folder contents, and modify.

However, only the administrator of the Web server should have these privileges to the Admin folder in which the Web Administrator pages are located. When secured in this way, if users attempt to access the Administrator page, the application prompts them for the machine’s administrator login ID and password.

In addition to the file-level security for the virtual directory and its contents, the Internet guest user also needs full control privileges to the Log folder in the MicroStrategy Common Files, located by default in `C:\Program Files (x86)\Common Files\MicroStrategy`. This ensures that any application errors that occur while a user is logged in can be written to the log files.

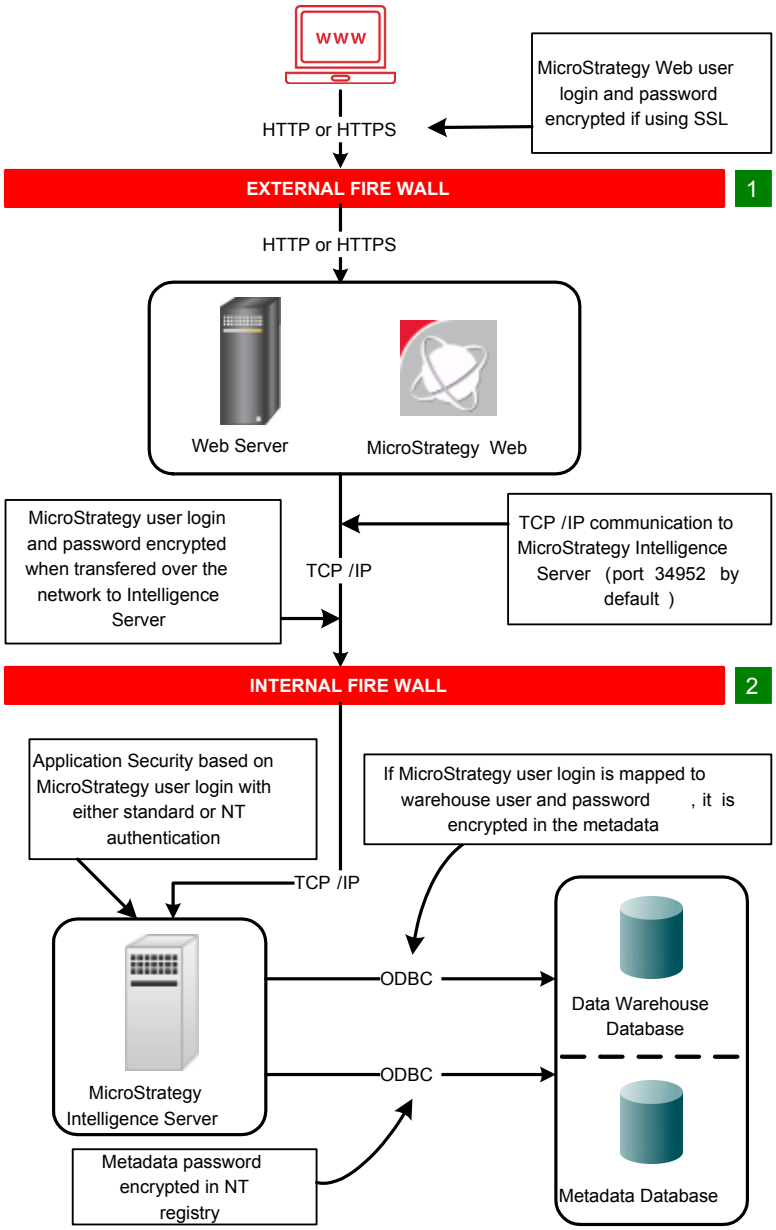
The file-level security described above is all taken care of for you when you install the ASP.NET version of MicroStrategy Web using Microsoft IIS. These details are just provided for your information.

If you are using the J2EE version of MicroStrategy Web Universal you may be using a different Web server, but most Web servers have similar security requirements. Consult

the documentation for your particular Web server for information about file-level security requirements.

Sample MicroStrategy system

The following diagram summarizes what a typical MicroStrategy system might look like if you take into account firewalls, digital certificates, and encryption:



 © 2016, MicroStrategy Inc.

Integrating Narrowcast Server with MicroStrategy Web products

It is possible to enable **Scheduled Delivery** and **Send Now** features in MicroStrategy Web products. The Scheduled delivery option allows users to have a report sent to an e-mail address that they specify on a certain schedule, to a printer or to a file location. These schedules are defined in MicroStrategy Narrowcast Server and are separate from the schedules maintained in Intelligence Server. The Send Now option allows users to send a report immediately to an e-mail address that they specify.

You must have MicroStrategy Narrowcast Server installed and configured before the Scheduled e-mail and Send Now options work. See that product's documentation for more information.

For more detailed information about this, see the *MicroStrategy Installation and Configuration Guide*.

To configure the Subscription Portal delivery option for MicroStrategy Web products, either the folder or the drive where the Subscription Engine is installed must be shared while the system is being configured. That is, the service running the Subscription Administrator must have read and write access to either:

- The folder where the Subscription Engine is installed

  or

- The entire drive

MicroStrategy Narrowcast Server and MicroStrategy Web products can automatically share this drive for the local Administrators group. The Subscription Administrator service should run under an account that is a member of the local Administrators group. You can unshare the drive or folder after the system is configured. If you do not want to automatically share the drive, perform the steps listed here.

To share the folder where the Subscription Engine is installed

1. Modify the Admin.properties file located on the Subscription Engine machine:

   ```
   ..\MicroStrategy\Narrowcast Server\Subscription
   Engine\build\server\
   ```

   Modify the file contents so the corresponding two lines are as follows:

   ```
   TransactionEngineLocation=machine_name:\\
   Subscription Engine\\build\\server
   ```

   ```
   TransactionEngineLocation=MACHINE_NAME:/Subscription
   Engine/build/server
   ```

© 2016, MicroStrategy Inc.

where *machine_name* is the name of the machine where the Subscription Engine is installed.

2. Share the folder where the Subscription Engine is installed for either the local Administrators group or for the account under which the Subscription Administrator service account runs. This folder must be shared as `Subscription Engine`.

- You should ensure that the password for this account does not expire.

If the Subscription Engine machine's drive is shared and unshared multiple times, the following Windows message displays: "`System Error: The network name was deleted.`"

- This message does not indicate a problem. Click **OK** to make the Subscription Administrator service functional.

3. Restart the Subscription Administrator service.

Enabling users to install MicroStrategy Office from Web

From the MicroStrategy Web Administrator page, you can designate the installation directory path to MicroStrategy Office, and also determine whether or not a link to Office installation information appears in the MicroStrategy Web interface. For steps to enable users to install MicroStrategy Office from Web, see the *Deploying MicroStrategy Web, Mobile Server, and Operations Manager* chapter in the *MicroStrategy Installation and Configuration Guide*.

FAQs for configuring and tuning MicroStrategy Web products

How do I configure my MicroStrategy Web environment if I have a user community of x users? How much hardware am I going to need?

This information is addressed in the MicroStrategy Knowledge Base.

© 2016, MicroStrategy Inc.

How do time-out settings in MicroStrategy Web and Intelligence Server affect MicroStrategy Web users?

Several settings related to session time-out may affect MicroStrategy Web users.

First, in the Intelligence Server Configuration Editor, under **Governing Rules: Default: General**, the value in the **Web user session idle time (sec)** field determines the number of seconds a user can remain idle before being logged out of Intelligence Server.

Second, in the `web.config` file, located by default in `C:\Program Files (x86)\MicroStrategy\Web ASPx`, the time-out setting determines the number of minutes after which the .NET session object is released if it has not been accessed. This time-out is independent of the Intelligence Server time-out above.

The section of the `web.config` file containing the time-out setting is as follows:

```
<sessionState
mode="InProc"
stateConnectionString=
"tcpip=127.0.0.1:42424"
sqlConnectionString="data source=
127.0.0.1;user id=sa;password="
cookieless="false"
timeout="20"
/>
```

This setting does not affect Web Universal because it does not use .NET architecture.

A third setting is the MicroStrategy Web Administration setting **Allow automatic login if session is lost**. This setting is in the Security section of the Web Administration page. It enables users to be automatically reconnected to Intelligence Server if the session is lost.

This setting does not automatically reconnect the .NET session object.

The following table demonstrates how the previous settings interact in various combinations.

Intelligence Server time-out	`web.config` time-out	Allow automatic login if session is lost	User idle time	Result
45 minutes	20 minutes	Either	30 minutes	User must log back in
20 minutes	45 minutes	No	30 minutes	User must log back in
20 minutes	45 minutes	Yes	30 minutes	User is automatically logged back in
20 minutes	45 minutes	Either	60 minutes	User must log back in

© 2016, MicroStrategy Inc.

A fourth group of settings is whether Web user sessions can be backed up and recovered. That is, if the user was viewing a report, document, or dashboard when the session was ended, when the user logs back in to Web, she can click a link to return to that report, document, or dashboard. If this is enabled, you can configure where and for how long the session is stored on disk. After the session is expired, the user cannot recover the session.

To configure these settings, access the Intelligence Server Configuration Editor, select the **Governing Rules: Default: Temporary Storage Settings** category. To enable the feature, select the **Enable Web User Session Recovery on Logout** check box, and in the **Session Recovery backup expiration (hrs)** field, type the number of hours you want to allow a session to be stored. In **Session Recovery and Deferred Inbox storage directory**, specify the folder where the user session information is stored.

How can I tune my MicroStrategy Web server for best performance?

- Clustering multiple web servers improves performance. For more information about this, see *Chapter 9, Clustering Multiple MicroStrategy Servers*.
- You can modify certain settings in the MicroStrategy Web server machine or application for best performance. Details for MicroStrategy Web and Web Universal follow:

MicroStrategy Web (ASP.NET)

The most significant things you can do:

- Tune Microsoft's Internet Information Services (IIS). For details, see the MicroStrategy Tech Notes TN11275 and TN7449.
- Increase the server machine's Java Virtual Machine heap size. For information on doing this, see MicroStrategy Tech Note TN6446.

MicroStrategy Web Universal (J2EE)

Tuning actions for the J2EE version of MicroStrategy Web Universal vary according to the Web server you are using. For tuning details, see the appropriate section in the *MicroStrategy Installation and Configuration Guide*.

Also, see the documentation for your particular Web application server for additional tuning information. In general, these are the things you can do:

- Use the MicroStrategy Web server instead of the application server to serve static files (such as CSS, JavaScript).
- Precompile JSPs according to the platform you are using.
- Increase the application server's Java Virtual Machine heap size.

 © 2016, MicroStrategy Inc.

14

COMBINING ADMINISTRATIVE TASKS WITH SYSTEM MANAGER

Introduction

MicroStrategy System Manager lets you combine multiple, sequential processes for your MicroStrategy environment into a single workflow that can be deployed at a scheduled time or on demand. You can create workflows for different tasks, such as installing, maintaining, and upgrading MicroStrategy environments; backing up projects; and launching or shutting down Cloud instances. These workflows can be deployed using a standard interface, an interactive command line process, or a completely silent configuration process.

- *Creating a workflow, page 546*: Includes steps to create a workflow using System Manager, as well as information on all the components required to create a workflow.
- *Defining processes, page 578*: Includes information on all the processes that can be included in a System Manager workflow. System Manager provides a set of MicroStrategy and non-MicroStrategy processes to include in a workflow.
- *Deploying a workflow, page 644*: Includes information on how to deploy a System Manager workflow. This includes deploying a workflow using a standard interface, an interactive command line process, and a completely silent configuration process, which is suited for OEM deployments.

© 2016, MicroStrategy Inc.

Creating a workflow

You use System Manager to create a workflow visually, by dragging and dropping processes and linking them together. This allows you to see the step-by-step process that leads the workflow from one process to the next. This visual approach to creating a workflow can help you to notice opportunities to troubleshoot and error check processes as part of a workflow.

The steps provided below show you how to create a workflow using System Manager. Additional details on the various components that constitute a System Manager workflow are provided after these steps.

Prerequisites

- It can be beneficial to determine the purpose of your workflow and plan the general logical order of the workflow before using System Manager.
- System Manager is installed. This tool is installed as part of the general MicroStrategy product suite.

To create a System Manager workflow

The steps provided below are expressed as a linear process. However, as you create a workflow, the steps of creating and modifying processes, connectors, decisions, parameters, and other components of a workflow can be interchanged as the requirements for a workflow are determined.

1. Open System Manager.
   - To open System Manager in a Windows environment:
     a. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **System Manager**.
   - To open System Manager in a UNIX or Linux environment:
     a. From a UNIX/Linux console window, browse to *HOME_PATH*, where *HOME_PATH* is the directory that you specified as the home directory during installation.
     b. Browse to the folder `bin`.
     c. Type `mstrsysmgrw`, and then press ENTER.

   The System Manager home page is displayed.
2. Select one of the following options to create a workflow:
   - **New Workflow**: Creates a new workflow. The steps below assume that you have selected this option to create a new workflow.

© 2016, MicroStrategy Inc.

- **Open Workflow**: Opens an existing workflow. You can create a new workflow based on an existing workflow.
- **Templates**: Opens a sample, template workflow for a configuration scenario. The sample workflows provide a framework for certain tasks, letting you modify the details for each process to work with the components and tools in your environment. For information on some of the available sample workflows, see *Sample workflows: Templates, page 571*.

An area to create a workflow in System Manager is displayed.

To create processes in a workflow

3 From the Connectors and processes pane, double-click a process to add it to the workflow. You can then move the process around so that it fits into the organization of the workflow.

By default, the first process created in a workflow is automatically defined as an entry process, and all other processes are automatically disabled as entry processes.

4 Right-click the process and select **Rename**. Type a new name for the process.

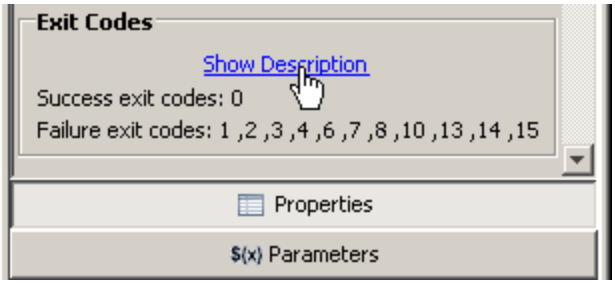
5 Select the process, and then select **Properties** in the pane on the right side. Provide all the required information for the process. For details on the properties required for each process, see *Defining processes, page 578*.

You can also use parameters to supply information for a process. To use a parameter in a process or decision, you must use the following syntax:

```
${ParameterName}
```

In the syntax listed above, `ParameterName` is the name of the parameter. During execution, this is replaced with the value for the parameter. Defining the parameters for a workflow is described in *To define the parameters for a process, page 548*, which is a part of this procedure.

6 While providing the information for a process, you can review the exit codes for a process. On the Properties pane, scroll down to the bottom and click **Show Description**, as shown in the image below.



Detailed information on each exit code for a process is displayed. For additional information on how you can use exit codes to help create a workflow, see *Determining process resolution using exit codes, page 637*.

© 2016, MicroStrategy Inc.

Exit code -4242424 is a general exit code that is shared among all processes. This exit code indicates that either the user canceled the workflow manually, or the reason for the process error cannot be determined.

7 Repeat the steps for *To create processes in a workflow, page 547* to create all the processes required for a workflow.

To define the parameters for a process

8 Each workflow has one set of parameters, which can be defined on the Parameters pane on the right side. The parameters can be used to provide values for a process when the workflow is executed. Using parameters can also let you provide this information in a secure fashion. For more information on how to include parameters in a workflow, including importing parameters from a file, see *Using parameters for processes, page 638*.

To define the logical order of a workflow

Once you have all the processes required for a workflow, you can begin to define the logical order of the workflow by creating connectors between all the processes. Each process in a workflow needs to connect to another process in the workflow, otherwise the workflow could end prematurely. You can also define a process as an entry processes of a workflow, create decisions to direct the logical order of a workflow, and add comments to provide further information and explanation to a workflow.

While defining the logical order of a workflow, you may find that additional processes are required. Processes can be added at any time while creating a workflow.

9 From the Connectors and processes pane, select from the following types of connectors:

- **Success**: The green arrow, to the left, is the success connector. If the process is completed with an exit code that is defined as a successful status, the process that the success connector points to is the next process that is attempted. If you use a success connector from a process, it is recommended that you also provide a failure connector.
- **Failure**: The red arrow, in the middle, is the failure connector. If the current process is completed with an exit code that is defined as a failure status, the process that the failure connector points to is the next process that is attempted. If you use a failure connector from a process, it is recommended that you also provide a success connector.
- **Continue**: The white arrow, to the right, is the continue connector. Regardless of the status of the exit code for the current process, the process that the continue

© 2016, MicroStrategy Inc.

connector points to is the next process that is attempted. If you use the continue connector from a process, you cannot use any other connectors for that process.

With a connector type selected, click the process to start from and drag to the process to proceed to next in the workflow. A connector is drawn between the two processes. If you use the success and failure connectors for a process, you must do this process for each connector.

These steps must be repeated for every process. For information on how to use connectors to define the logical order of a workflow, see *Using connectors to create the logical order of a workflow, page 553*.

10 From the Connectors and processes pane, select the **Decision** icon, and then click in the workflow area. A decision process is created in the workflow, as shown in the image below.

Decisions provide the ability to determine the next process in a workflow based on specific exit codes for the previous process, rather than just the simple success or failure of a process. For examples of how decisions can be used to define the logical order of a workflow, see *Using decisions to determine the next step in a workflow, page 555*.

Create as many decisions as you need for your workflow. Each decision should use a success and a failure connector to other processes in the workflow.

11 To enable or disable a process as an entry process for the workflow, in the workflow area, right-click the process and select **Toggle Entry Process**.

An entry process is a process that can be selected as the first process to attempt when the workflow is executed. For information on how to use entry processes, see *Using entry processes to determine the first step in a workflow, page 554*.

A process that is defined as an entry process is displayed with a green flag symbol, as shown in the image below.

By default, the first process created in a workflow is defined as an entry process, and all other processes are disabled as entry processes.

12 To process related tasks one by one, from the Connectors and processes pane, select the **Iterative Retrieval** icon, and then click in the workflow area. An iterative retrieval process is created in the workflow, as shown in the image below.

© 2016, MicroStrategy Inc.

With an iterative retrieval process, you can have a workflow retrieve information from sources including a folder, the contents of a file, or a System Manager parameter. This information can then be passed to another process in the System Manager workflow for processing a task. For example, by using an iterative retrieval process, a folder that stores weekly update packages can be analyzed to determine how many update packages need to be applied for a week, and then apply these updates one by one.

For information on how you can use the iterative retrieval process to perform related tasks one by one in a workflow, see *Processing related tasks one by one, page 560*.

13 To create a split execution process, from the Connectors and processes pane, select the **Split Execution** icon, and then click in the workflow area. A split execution process is created in the workflow, as shown in the image below.

A split execution process lets you start multiple threads in a workflow to perform parallel processing of the tasks. This can speed up a workflow for systems that can handle the parallel processing.

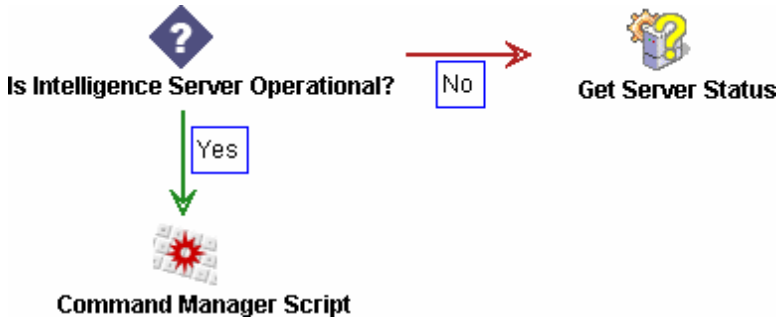
As tasks are executed simultaneously, you can also determine if additional tasks in the workflow should be processed if tasks that were performed in parallel are completed. From the Connectors and processes pane, select the **Merge Execution** icon, and then click in the workflow area. A merge execution process is created in the workflow, as shown in the image below.

For information on how you can use the split execution and merge execution to handle the parallel processing of tasks in a workflow, see *Processing multiple tasks simultaneously, page 562*.

14 To create a comment in the workflow, from the Connectors and processes pane, select the **Comment** icon, and then click in the workflow area.

Comments can be used to explain the design of a workflow. For example, you can use comments to explain the paths of a decision process, as shown in the image below.

© 2016, MicroStrategy Inc.



You can add as many comments as needed to explain a workflow. Be aware that the comments are viewable only in System Manager and cannot be displayed to a user while the workflow is being executed. For information on how to use comments to add context to a workflow, see *Using comments to provide context and information to a workflow, page 567*.

To define the end of a workflow

15 Create an exit process, which ends the workflow and can explain how the workflow ended. From the Connectors and processes pane, select the **Exit Workflow** icon, and then click in the workflow area. An exit process is created in the workflow, as shown in the image below.

16 With the exit process selected, from the Properties pane, you can choose to have the exit process return the exit code from the previous process or return a customized exit code. For more information on how to use exit processes to end a workflow, see *Using exit processes to end a workflow, page 559*.

17 Create connectors from any processes that should be followed by ending the workflow. All processes should lead to another process in the workflow, a decision, or an exit process.

To validate a workflow

18 From the **Workflow** menu, select **Validate Workflow**. One of the following messages is displayed:

- If the workflow is listed as valid, click **OK** to close the dialog box.
- If the workflow is not valid, click **Details** to review the reasons why the workflow is not valid. Click **OK** and make any required changes to the workflow. Once all changes are made, validate the workflow again.

For information on what is checked when validating a workflow, see *Validating a workflow, page 568*.

© 2016, MicroStrategy Inc.

To save a workflow

19 From the **File** menu, select **Save Workflow As**. The Save As dialog box opens.

20 Type a name for the workflow and select a location to save the workflow, and then click **Save**.

To deploy a workflow

The steps below show you how to deploy a workflow from within System Manager. For information on deploying a workflow from the command line or as a silent process, see *Deploying a workflow, page 644*.

21 From the **View** menu, select **Options**. The Options dialog box opens.

22 In the **Log file path** field, type the path of a log file, or use the folder (browse) icon to browse to a log file. All results of deploying a workflow are saved to the file that you select.

23 Click **OK** to save your log file selection and close the Options dialog box.

24 From the **Workflow** menu, point to **Execute Workflow**, and then click **Run Configuration**. The Run configuration dialog box opens.

25 From the **Starting process** drop-down list, select the process to act as the first process in the workflow. You can select only a process that has been enabled as an entry process for the workflow.

26 In the **Parameters** area, type any parameters required to execute the processes in the workflow, which may include user names, passwords, and other values. To include multiple parameter and value pairs, you must enclose each parameter in double quotes (`" "`) and separate each parameter and value pair with a space. For example, `"UserName=User1" "Password=1234"` is valid syntax to provide values for the parameters `UserName` and `Password`.

For information on supplying parameters for a workflow, see *Using parameters for processes, page 638*.

27 Click **Run** to begin the workflow. As the workflow is being executed, the results of each process are displayed in the Console pane. The results are also saved to the log file that you specified earlier.

If you need to end the workflow prematurely, from the **Workflow** menu, select **Terminate Execution**. A dialog box is displayed asking you to verify your choice to terminate the execution of the workflow. Click **Yes** to terminate the workflow. If some processes in the workflow have already been completed, those processes are not rolled back.

© 2016, MicroStrategy Inc.

Using connectors to create the logical order of a workflow

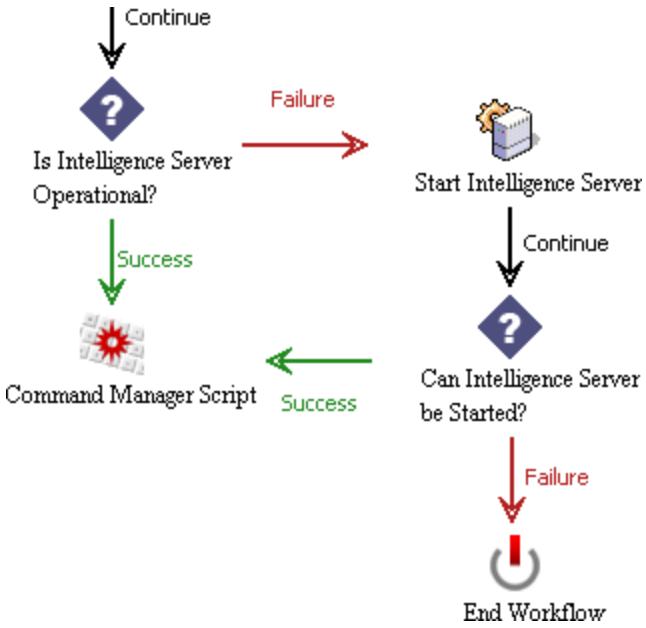
When a process in a workflow is completed, the next step to take in a workflow is determined using connectors. *Connectors* determine the logical order of a workflow according to the exit code of the process they are coming from. You can select from the following types of connectors:

- **Success**: The green arrow, to the left, is the success connector. If a process is completed with an exit code that is defined as a successful status, the process that the success connector points to is the next process that is attempted. If you use a success connector from a process, it is recommended that you also provide a failure connector. Without a failure connector, the workflow may unexpectedly end with the current process.
- **Failure**: The red arrow, in the middle, is the failure connector. If a process is completed with an exit code that is defined as a failure status, the process that the failure connector points to is the next process that is attempted. If you use a failure connector from a process, it is recommended that you also provide a success connector. Without a success connector, the workflow may unexpectedly end with the current process.
- **Continue**: The white arrow, to the right, is the continue connector. Regardless of the status of the exit code for a process, the process that the continue connector points to is the next process that is attempted. If you use the continue connector from a process, you cannot use any other connectors for that process.

When a connector is added to a workflow, it is drawn from one process to another. The arrow for the connector points to the next process to attempt in a workflow, and the start of the connector links to the process that was just completed.

It is common to use a combination of success and failure connectors to lead from a process. These connectors allow you to continue with the main workflow if the process was successful, and end the workflow or troubleshoot the problem if the process was unsuccessful. For example, the steps of a workflow shown in the image below show success and failure connectors leading from a decision process.

© 2016, MicroStrategy Inc.



The first decision process shown in the image above determines if Intelligence Server is operational. If so, the workflow follows the success connector to continue on to a Command Manager script to perform various configurations. If Intelligence Server is not operational, the workflow follows the failure connector on an alternative path to attempt to start Intelligence Server before attempting the Command Manager script.

This example also includes a few continue connectors. For example, the Start Intelligence Server process uses a continue connector to lead to a decision process. The decision process is then used to determine the exit code of the previous process. For examples of how decisions can be used to define the logical order of a workflow, see *Using decisions to determine the next step in a workflow, page 555*.

Using entry processes to determine the first step in a workflow

When you deploy a workflow, you can choose which process is the first to attempt in a workflow. This allows you to skip steps that have already been accomplished or are not required in certain environments. Being able to select the process to begin with can also be helpful when creating a workflow as part of testing and troubleshooting the steps in a workflow.

An *entry process* is any process in a workflow that can be selected as the first process to attempt in a workflow. You can enable and disable processes in a workflow as available entry processes for the workflow. By default, the first process created in a workflow is defined as an entry process; all other processes are disabled as entry processes.

To be able to select a process in a workflow as the first process to attempt, it must be enabled as an entry process. In the workflow area, right-click a process and select **Toggle Entry Process**. This enables or disables a process as an entry process for the workflow. A process that is defined as an entry process is displayed with a green flag symbol, as shown in the image below.

 © 2016, MicroStrategy Inc.

Although any process, other than an exit process, can be enabled as an entry process for a workflow, you should limit the steps that are enabled as entry processes for various reasons:

- Some steps in a workflow may not work as entry processes. For example, a decision process that relies on the exit code of the previous process should not be enabled as an entry process. This is because the decision process could not retrieve the required exit code. Without the ability to retrieve an exit code, the decision process would not be able to perform a comparison, and the workflow would appear to be unresponsive.
- When deploying a workflow using System Manager, each available entry process is listed. Providing many available entry processes can cause confusion as to which entry process to use to begin the workflow.
- When deploying a workflow, starting at a certain step can cause previous steps to be skipped entirely, depending on the logical order of the workflow. Ensure that skipping certain steps still allows the workflow to be valid in the scenarios that it will be used in.

Using decisions to determine the next step in a workflow

When a process is completed, the simple success or failure of a process is not always enough to determine the next step to take in a workflow. Decision processes can be used to compare process exit codes, parameters, and other values to provide additional control over the next step to take in a workflow. You can also use decision processes to check for the existence of a file or folder, as well as if the file or folder is empty.

To add a decision process to your workflow, from the Connectors and processes pane, select the **Decision** icon, and then click in the workflow area. A decision process is created in the workflow, as shown in the image below.

Once a decision process is added to a workflow, from the Properties pane, you can define the comparison for the decision process as follows:

- *To compare parameters, constants, and exit codes, page 556*
- *To check for the existence of a file or folder, page 556*

To compare parameters, constants, and exit codes

1 Select the option **Parameter/Exit Code Comparison**.

2 Select to use a parameter or an exit code as the first item for the comparison:

- **Parameter or constant**: Select this option to provide a parameter or constant for comparison. You must type the parameter name or the constant value.
- **Previous process exit code**: Select this option to use the exit code of the previous process in the comparison. Using the exit code of a process allows you to determine in greater detail why a process was successful or unsuccessful. This allows you to take more specific action to troubleshoot potential problems in a workflow.

For example, if you attempt to execute a Command Manager script as part of a workflow, this type of process can fail for various reasons. If the process fails with an exit code equal to four, this indicates that a connection could not be made to perform the script. For this exit code, a decision process could lead to a process to start Intelligence Server. However, if the process fails with an exit code equal to six, this indicates that the script has a syntax error. For this exit code, a decision process could lead to an exit process, so the workflow could be ended and the Command Manager script could be manually reviewed for syntax errors.

3 From the **Comparison operator** drop-down list, select the operator for the comparison. This includes standard operators such as Equal to, Not equal to, Greater than, and so on.

4 In the **Comparison item 2** field, type a value. It is common to type a constant value to compare a parameter or exit code to.

5 In the Output parameters area, you can specify a parameter in the **Previous process exit code** drop-down list. The parameter specified is updated with the value of the exit code from the process that was completed just before the decision process. You can use this technique if you need multiple decision processes to determine the next course of action, which is described in *Using multiple decision processes to troubleshoot a workflow , page 557* below.

If you do not need to use the exit code from the previous process later in the workflow, you can leave the Previous process exit code drop-down list blank.

To check for the existence of a file or folder

1 Select the option **File/Directory Check**.

2 In **File/Directory Path**, type the path to the file or directory to check. You can also click the folder icon to browse to and select a file or directory.

3 From the **File/Directory Check Condition** drop-down list, select one of the following options:

 © 2016, MicroStrategy Inc.

- **Exists**: Select this option to check only if the file or directory exists. The decision process returns as true if the file or directory can be found.
- **Exists and not empty**: Select this option to check if the file or directory exists, and if the file or directory is empty. For files, this check verifies that some information is in the file. For directories, this check verifies whether any other files or folders are in the directory. The decision process returns as true if the file or directory exists, and the file or directory has some type of content available.

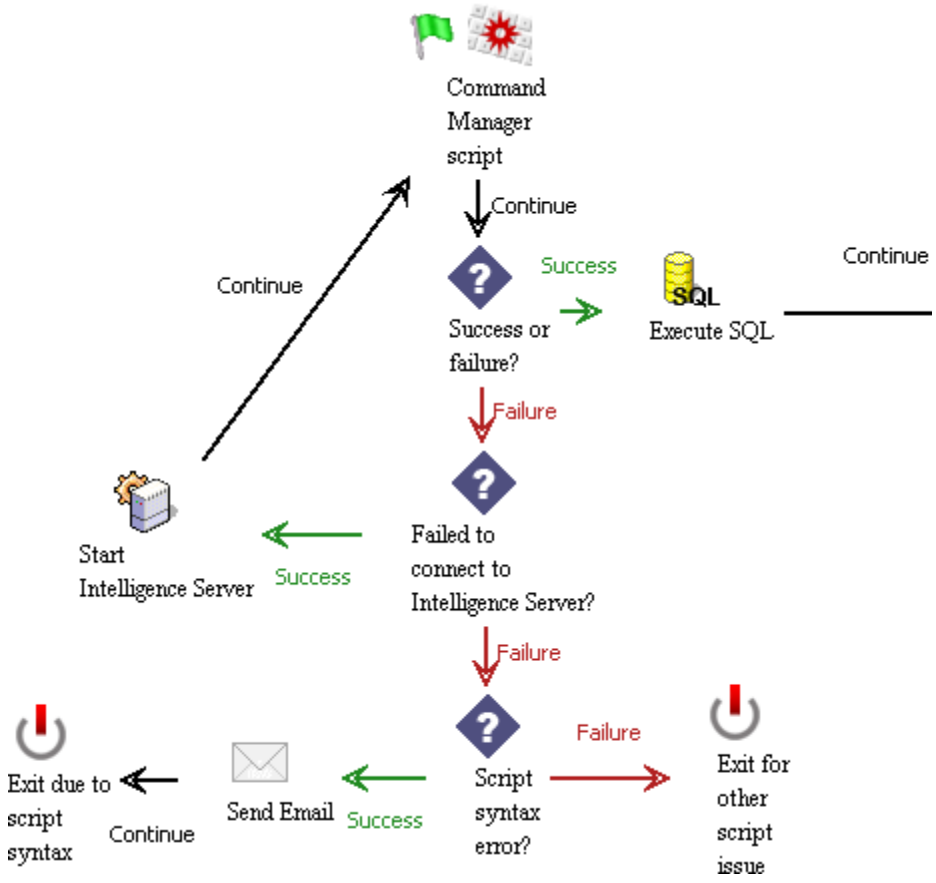
Using multiple decision processes to troubleshoot a workflow

When you are creating a workflow, you can use multiple decision processes to take more specific action on process exit codes and troubleshoot potential problems in a workflow.

When a process in a workflow is completed, it can either be a success or failure. Additionally, certain processes can fail for multiple reasons. Although a single decision process can determine if a process was a success or failure, you need to use multiple decision processes to qualify how a process failed. By qualifying why a process failed, you can more accurately troubleshoot the process and, in some cases, even take action in the workflow itself to fix the problem.

For example, if you attempt to execute a Command Manager script as part of a workflow, this type of process can fail for various reasons. If the process fails with an exit code equal to four, this indicates that a connection could not be made to perform the script. For this exit code, a decision process could lead to a process to start Intelligence Server. However, if the process fails with an exit code equal to six, this indicates that the script has a syntax error. For this exit code, a decision process could lead to an exit process, so the workflow could be ended and the Command Manager script could be manually reviewed for syntax errors. T troubleshooting scenario is shown in the workflow below.

© 2016, MicroStrategy Inc.



The first decision process (labeled as Success or failure?) can determine whether the Command Manager script was a success or a failure. Additionally, this decision process uses the Previous process exit code to store the exit code for the Command Manager script process into a parameter called `Decision`. You must use the Previous process exit code to store the exit code for the original Command Manager process so that this exit code can be used in the other decision processes.

In a chain of multiple-decision processes, you should use the Previous process exit code option only in the first decision process. This is because once this exit code is stored in a parameter, you can then reuse that parameter in later decision processes as a comparison item. If you were to mistakenly include the same parameter in the Previous process exit code option for one of the later decision processes, the parameter would be updated to have the exit code of the previous decision process. This would then overwrite the original exit code, which would prevent you from comparing the original exit code in the later decision processes.

If the script was a success, the first decision process allows the workflow to continue. If the script fails, a second decision process is started. This second decision process (labeled as Failed to connect to Intelligence Server?) uses the value previously stored in the `Decision` parameter to determine if the exit code is equal to four. With an exit code equal to four, this decision process can attempt to start Intelligence Server and then attempt to run the Command Manager script again. If this second decision process fails, which means the exit code is not equal to four, a third decision process (labeled as Script syntax error?) is started.

 © 2016, MicroStrategy Inc.

This third decision process again uses the value that was stored in the `Decision` parameter by the first decision process to determine if the exit code is equal to six. With an exit code equal to six, this decision process can send an email to a someone to review the Command Manager script for syntax errors, and it can attach the script to the email. Once the email is sent, the workflow is exited. If this final decision process fails, that means the Command Manager script failed for another reason. In this case, the workflow is exited for additional troubleshooting.

When using multiple decision processes to qualify the resolution of a previous process, be aware that as long as you store the original exit code in a parameter, you can use as many decision processes as necessary.

Additionally, this technique of using multiple decision processes is a good practice for processes that are important to the overall success or failure of a workflow. However, using this technique for every process in a workflow could cause the workflow to become overly complex and difficult to create and follow. For example, processes that send emails likely do not require involved troubleshooting in the workflow itself, but a process that attempts to start Intelligence Server may benefit from including potential troubleshooting steps.

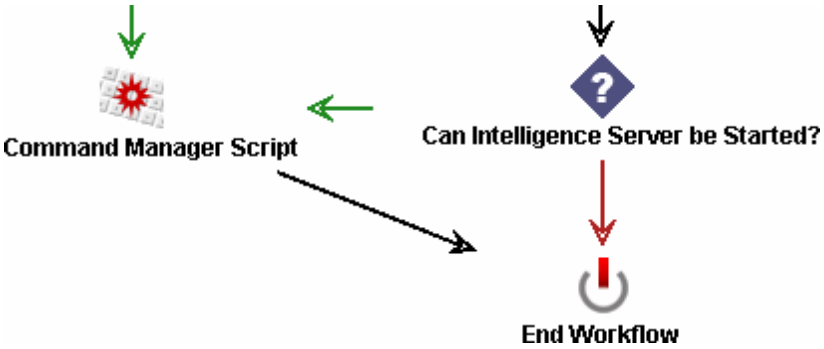
Using exit processes to end a workflow

When a workflow is deployed, it is important to be able to notify whoever is deploying the workflow when and how the workflow has ended. An *exit process* allows you to end a workflow and explain how the workflow ended.

To add an exit process to your workflow, from the Connectors and processes pane, select the **Exit Workflow** icon, and then click in the workflow area. An exit process is created in the workflow, as shown in the image below.

With the process selected, from the Properties pane, you can define what type of exit code is provided when the exit code is reached:

- **Use previous process exit code**: Select this option to return the exit code of the process that was completed just before the exit process. If you use this option you can use the same exit process from multiple processes in the workflow, and the exit code returned provides information on whatever process led to the exit process. For example, the steps of a workflow shown in the image below show two processes leading to the same exit process.



When the workflow completes, the same exit process returns the exit code either on the decision process that determines if Intelligence Server can be started, or the process that completes a Command Manager script.

- **Use customized exit code**: Select this option to define your own exit code for the exit process by typing in the available field. This allows you to create exit codes customized to your needs. You can use only numeric values for the customized exit code.

  If you use this option, you may want to use multiple exit processes in a workflow. You can then define each exit process with a unique exit code. This can explain what path the workflow took and how it ended. This can be helpful because workflows can have multiple possible paths including a successful path where all processes were completed and unsuccessful paths where the workflow had to be ended prematurely.

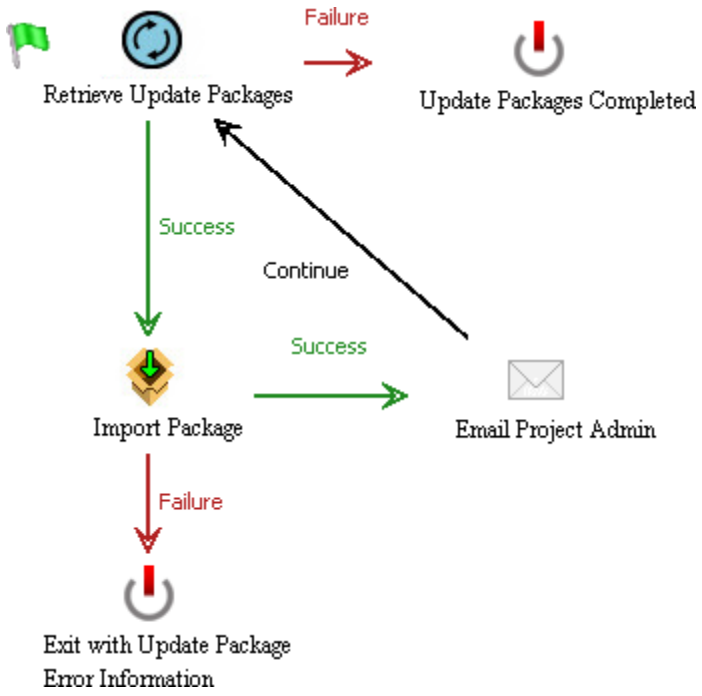
Every workflow should include at least one exit process. Ensuring that processes either lead to another process or to an exit process provides a consistent expectation for the results of a workflow.

Processing related tasks one by one

System Manager supports processing related tasks one by one and determining how many related tasks are available. This can be done using the iterative retrieval process. With such a process, you can have a workflow retrieve information from sources including a folder, the contents of a file, or a System Manager parameter. This information can then be passed to another process in the System Manager workflow for processing a task.

For example, you have multiple projects that require object updates on an intermittent schedule. At the start of each week, any updates that are required are included in a separate update package for each project, and all update package files are stored in a folder. The number of update packages required for a week varies depending on requirements of the various projects. By using the iterative retrieval process, the folder that stores the weekly update packages can be analyzed to determine how many update packages need to be applied for the week. The workflow shown below then retrieves these update packages from the folder one by one, applying the update package, emailing the project administrator, and using the iterative retrieval process to retrieve the next update package.

 © 2016, MicroStrategy Inc.



The iterative retrieval process automatically determines the number of update packages in the folder, which allows you to run the same workflow each week without having to modify the workflow to account for varying numbers of update packages from week to week. Once all update packages are processed and no more update packages can be retrieved, the iterative retrieval process exits with a failure exit code to signify that no more information is available for retrieval.

With the process selected, from the Properties pane, you can define how the iterative retrieval process retrieves information to be processed as part of a System Manager workflow:

- **Files in Directory**: Select this option to retrieve files from a folder. When retrieving files from a folder, be aware that each time a file is retrieved, it is stored in the same parameter and thus provided to the same process in the System Manager workflow. This means that the System Manager process that uses these files must be able to process all files in a folder. In the example update package scenario, the folder must contain only update packages. If, for example, a text file was stored in the folder, retrieving this text file and passing it to the import package process would cause an error in the workflow.

  Click the folder icon to browse to and select a folder, or type the full path in the **Directory Name** field. You must also determine how the files are retrieved, using the following options:

  - **File Names Only**: Select this option to retrieve only the name of the file, including the file extension. If you clear this check box, the full file path to the file is retrieved, which is commonly required if you need the location of the file for other processes in the System Manager workflow.
  - **All Files**: Select this option to retrieve files from only the top-level folder.

    - **All Files and Subfolders Recursively**: Select this option to retrieve files from the top-level folder and all subfolders.

- **Content of File**: Select this option to retrieve the contents of a file. Click the folder icon to browse to and select a file, or type the full path in the **File Name** field. You must also determine if a separator is used to segment the content within the file, using the following option:

    - **Separator**: Select this check box to retrieve the contents of a file in multiple, separate segments. Type the separator character or characters that are used in the file to denote separate sections of content. For example, you can type a comma (`,`) if the content is separated using commas. You can also use characters such as `\n`, `\t`, and `\s` to represent the new line, tab, and space separators, respectively.

      If you clear this check box, the entire contents of the file is returned in a single retrieval.

- **Parameter**: Select this option to retrieve the contents of a parameter. From the **Parameter Name** drop-down list, select a parameter that is included in the System Manager workflow. You must also determine if a separator is used to segment the content within the parameter, using the following option:

    - **Separator**: Select this check box to retrieve the contents of a parameter in multiple, separate segments. Type the separator character or characters that are used in the parameter to denote separate sections of content. For example, you can type a comma (`,`) if the content is separated using commas. You can also use the characters `\n`, `\t`, and `\s` to represent the new line, tab, and space separators, respectively.

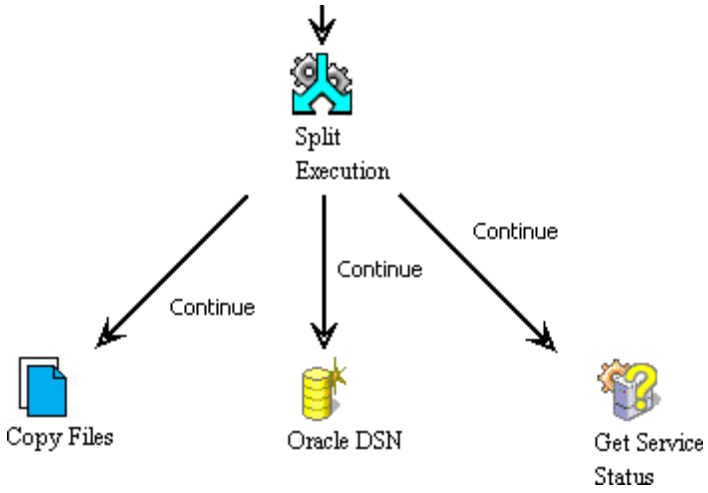
      If you clear this check box, the entire contents of the parameter is returned in a single retrieval.

- **Output Parameter**: The information retrieved must be stored in a parameter so that it can be passed to another process in the System Manager workflow. Select an output parameter from the drop-down list.

Processing multiple tasks simultaneously

System Manager supports executing tasks in parallel in a workflow. This takes advantage of a system's processing power to complete the tasks more quickly. This is done by using the split execution process. With a split execution process, you can have a workflow process two or more tasks at the same time. The split execution process shown below takes a linear workflow and begins to process three tasks in parallel.

 © 2016, MicroStrategy Inc.

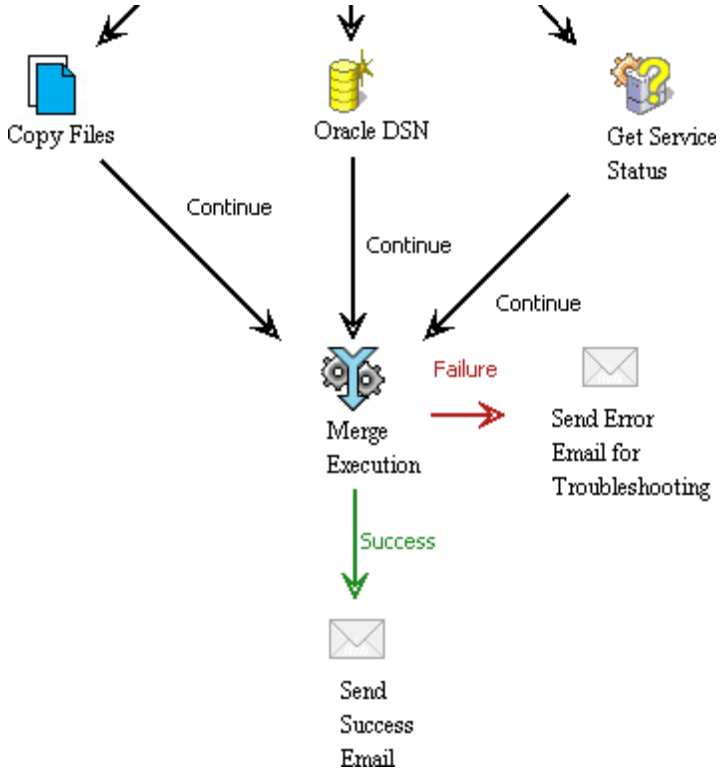


When using split executions to process multiple tasks in a workflow at the same time, consider the following best practices:

- Ensure that the tasks do not depend on each other. Workflows are often linear processes that require that one task is completed before starting another task. For example, you cannot run certain Command Manager scripts until Intelligence Server is started. This means a task to start Intelligence Server should not be done in parallel with other tasks that require Intelligence Server to be operational.
- Consider the amount of processing that is required to perform the tasks in parallel, relative to your available system resources. While performing multiple tasks at once can save time, it can also slow down overall performance if the required system resources are not available. Even if a workflow is created to start multiple tasks, you can limit the number of tasks that are performed in parallel to prevent overloading the system, as described in *Limiting the number of parallel tasks to prevent over consumption of system resources, page 565*.
- Split execution processes can use only the continue connector (see *Using connectors to create the logical order of a workflow, page 553*) to link to new tasks to perform in parallel. You must also use two or more continue connectors, as a split execution is meant to split a workflow into at least two paths to perform in parallel.

Once a workflow execution is split into multiple paths, each task is performed independently of the other tasks. However, while the tasks are done independently, all the tasks may need to be completed before performing other tasks later in the workflow. For example, you can create a DSN and start Intelligence Server as separate tasks at the same time, but you may need both of those tasks to be fully complete before starting another task that requires the DSN to be available and Intelligence Server to be operational. To support this workflow, you can use the merge execution process to combine multiple paths back into one workflow path. For example, the merge execution process shown below combines the three tasks performed in parallel back into one execution after the three tasks are completed.

© 2016, MicroStrategy Inc.

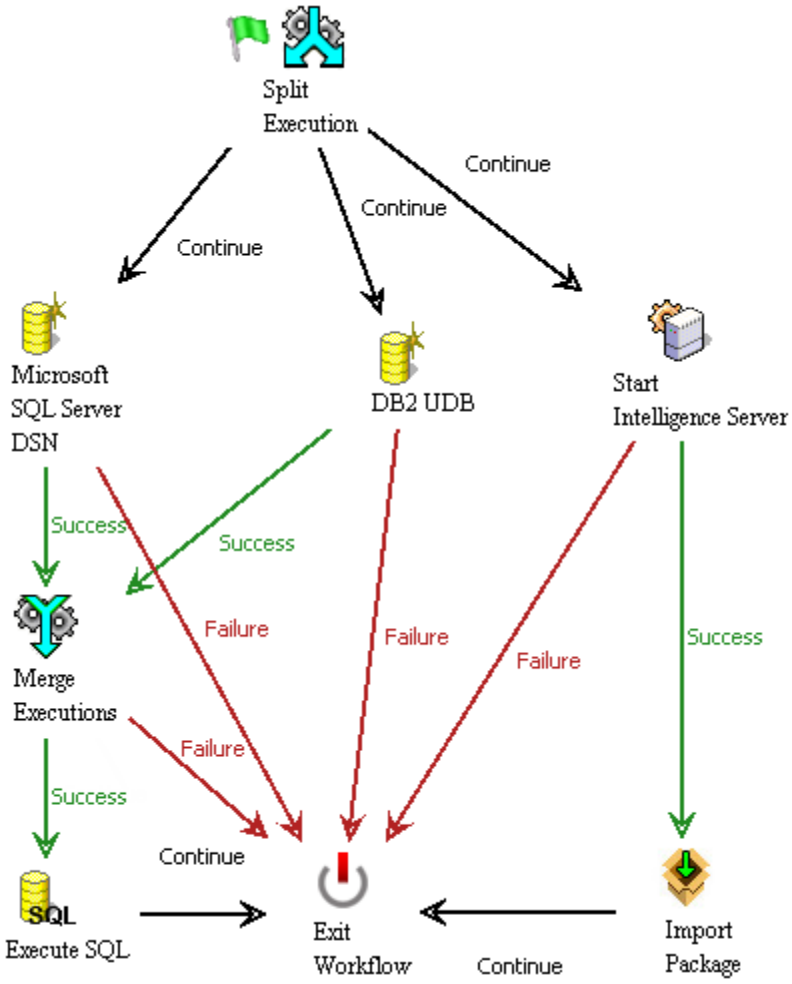


For each merge execution process, you must supply a time out value. This time out value is the amount of time, in seconds, that is allowed to complete all the parallel tasks that are connected to the merge execution process. The time starts to count down once the first task connected to a merge execution process is completed. How the remaining tasks connected to the merge execution are processed depends on the connectors used to continue from the merge execution process:

- It is recommended that you use the success and failure connectors to exit the merge process:
  - Success connector: If each task that is connected to a merge execution is completed in the allotted time, the workflow continues to the configuration that is linked to the merge execution with the success connector.
  - Failure connector: If at least one task connected to the merge execution is not completed in the allotted time, or all other paths have been ended without reaching the merge execution process, the workflow continues to the configuration that is linked to the merge execution with the failure connector.

Although merge execution processes are helpful to continue the workflow when certain tasks are completed, you do not have to merge any or all paths that are started with a split execution process. Each task performed in parallel with other tasks can come to separate completions using standard exit processes (see *Using exit processes to end a workflow, page 559*). For example, in the workflow shown below, both DSN creation configurations must be completed to also process the Execute SQL configuration. However, the path that starts with an Intelligence Server startup configuration continues on to completion regardless of whether any of the other tasks are completed.

 © 2016, MicroStrategy Inc.



Limiting the number of parallel tasks to prevent over consumption of system resources

While creating a workflow, the split execution process can be used to start as many tasks at the same time as required. However, each additional task that is attempted in parallel requires additional system resources. If your system cannot handle the additional processing requirements to complete all the tasks in parallel, this can slow down the workflow execution and the entire system's performance.

To avoid these types of performance issues, you can limit the number of tasks that can be processed at the same time for a workflow. This ensures that even if a workflow requests a certain number of tasks to be processed at the same time, only the specified limit is allowed to run at a time.

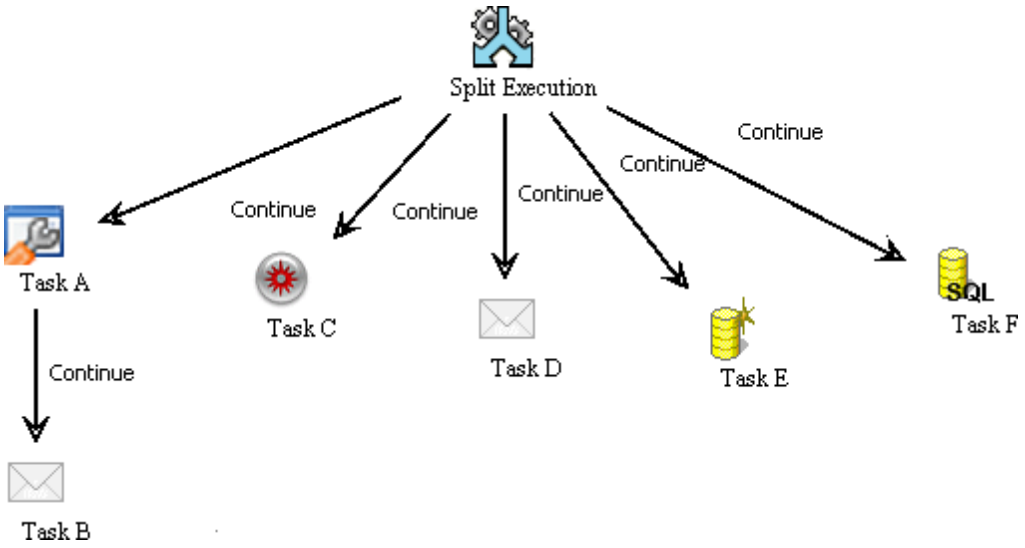
The default value for the limit is the greater of either the number of CPUs for the system or 2. Although the number of CPUs for the system is a reasonable default, be aware of the following:

- Systems can process more tasks simultaneously than the number of CPUs available.

- Systems can have multiple CPUs, but this does not necessarily mean all the CPUs are available to the user who is deploying a workflow. For example, consider a Linux machine with eight CPUs available. In this scenario, the Maximum Thread default value is 8. However, the user account that is being used to deploy the workflow may be allowed to use only one CPU for the Linux machine. When determining the maximum number of tasks to run simultaneously in System Manager workflows, you should understand details about system resource configuration.

As a workflow is deployed, any tasks over the set limit are put into a queue. For example, if a split execution process attempts to start five tasks, but the Maximum Threads option is set at three, two of the tasks are immediately put in the queue. Once a task is completed, the next task in the queue can begin processing.

In terms of queueing and processing tasks, each separate configuration is considered as a separate task. Once a configuration is completed, the configuration that it links to next might not be the next configuration to be processed. For example, a split execution process attempts to start five tasks, as shown in the image below.



The Maximum Threads option is set at three, which means that two of the tasks are immediately put in the queue. Assume then that one of the three tasks being processed (Task A) comes to completion, and it links to another task in the workflow (Task B). Rather than immediately starting to process Task B, the workflow must first process the tasks that were already included in the queue (Task E and Task F). This puts Task B behind the two existing tasks already in the queue.

To define the parallel task limit

1. From the **View** menu, select **Options**. The Options dialog box opens.
2. In the **Maximum Concurrent Threads** field, type the maximum number of tasks that can processed at the same time. The default value for this option is the greater of either the number of CPUs for the system or 2.
3. Click **OK**.

 © 2016, MicroStrategy Inc.

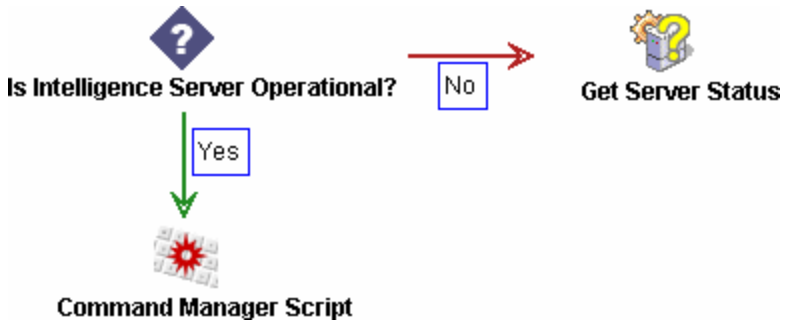
Using comments to provide context and information to a workflow

Workflows can be made more helpful if you add information about why certain steps are performed or explain the logical order of the workflow. You can include this type of information in a workflow by adding comments to the workflow.

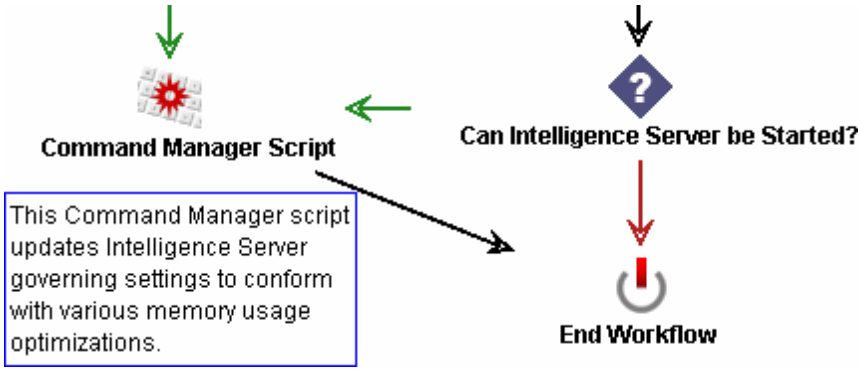
To add a comment to your workflow, from the Connectors and processes pane, select the **Comment** icon, and then click in the workflow area. A comment is created in the workflow.

You can then type the information for the comment. You can also resize the comment and move it to the required location in a workflow.

You can use comments to explain to the workflow's design. For example, you can use comments to explain the paths of a decision process, as shown in the image below.



Another benefit of using comments is to provide information directly in the workflow area. For example, the image below shows a workflow with a comment that explains the Command Manager script process.



The same information in the comment is included in the description for the Command Manager script process. However, providing the information in a comment allows this information to be displayed directly in the workflow area.

You can add as many comments as needed to explain a workflow. Be aware that the comments are viewable only in System Manager and cannot be displayed to a user while the workflow is being deployed.

Validating a workflow

Validating a workflow is an important step in creating a workflow. Although validating a workflow does not guarantee that every process will be completed successfully when deploying a workflow, it helps to limit the possibility for errors during the deployment.

While you are creating a workflow, you can use System Manager to validate the workflow. This validation process performs the following checks on the workflow:

- The workflow contains at least one entry process. This is required so that the workflow has at least one process to use as the first step in the workflow.
- All processes have values for all required properties. For example, if you are creating a DSN, you must supply a name for the DSN, the machine that stores the data source, the port number, and other required values for the data source type.

  The validation checks only that values exist for all required properties, not whether the values are valid for the process.

To use System Manager to validate a workflow, from the **Workflow** menu, select **Validate Workflow**. One of the following messages is displayed:

- **The workflow is valid**: This message is displayed if it passes all the checks listed above. Click **OK** to close the message.
- **Incomplete workflow**: This message is displayed if parts of the workflow are not valid. Click **Details** to display the invalid portions of the workflow. Continue to fix the workflow and perform the validation until you see the message, "The workflow is valid." Click **OK** to close the message.

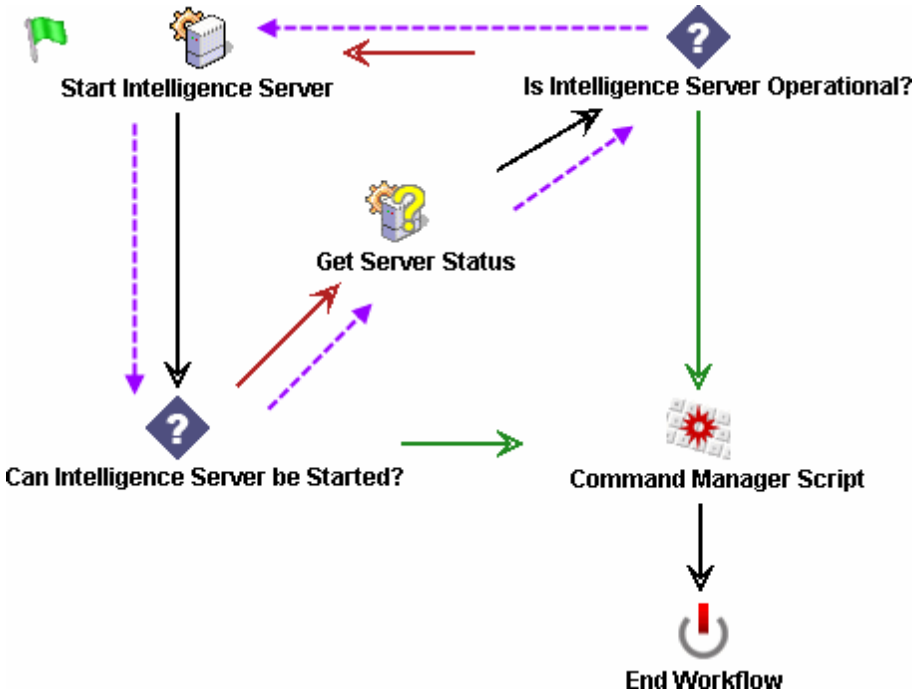
Additional validations that can be done manually on a workflow are described in *Manually validating a workflow, page 568* below.

Manually validating a workflow

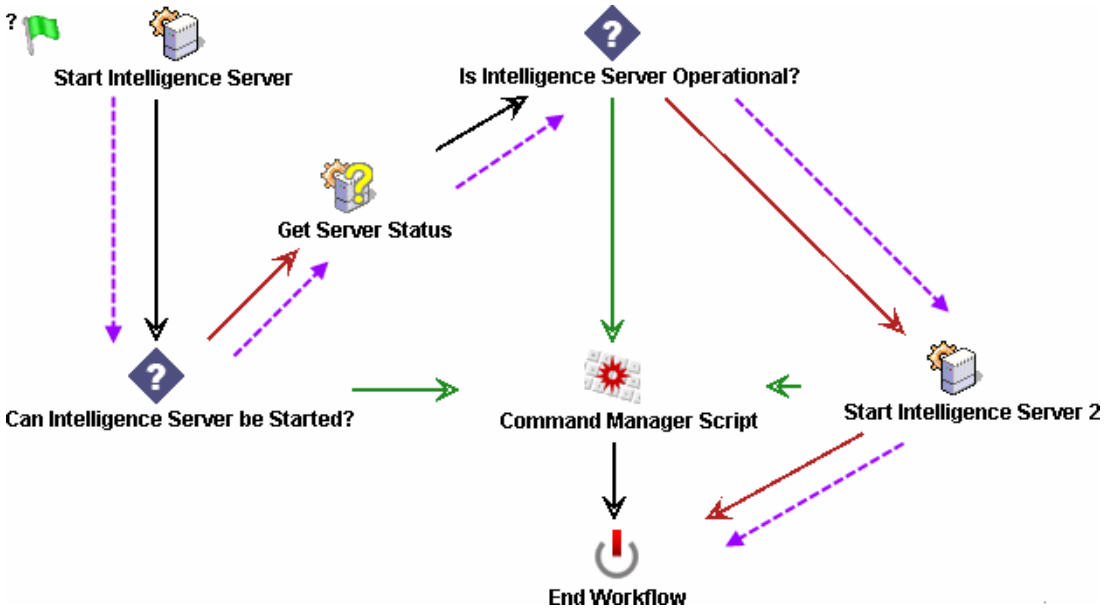
As part of validating your workflow, you should manually validate additional aspects of the workflow. These additional validations help reduce the potential for issues to develop when deploying a workflow. This includes the following validations:

- Each process has either one continue connector or one success connector and one failure connector leading from it. This ensures that each process continues on to another step in the workflow regardless of whether the process is successful or unsuccessful. For more information on correctly supplying connectors for a workflow, see *Using connectors to create the logical order of a workflow, page 553*.
- The workflow has at least one exit process. Exit processes verify that a workflow deployment has completed. For more information on how you can use exit processes in a workflow, see *Using exit processes to end a workflow, page 559*.
- Step through the logical order of the workflow and double-check that all the possible paths make sense with the purpose of the workflow. You can also use this as an opportunity to check for parts of the workflow that could become cyclical. For example, in the workflow shown in the image below, a potential cyclical path is highlighted with purple, dashed arrows.

 © 2016, MicroStrategy Inc.



Although this cyclical path would let the workflow attempt to start Intelligence Server multiple times, if Intelligence Server cannot be started successfully, the workflow could continue to execute until it was manually ended. An alternative would be to modify the logical order of the workflow to attempt to start Intelligence Server a second time, but end the workflow if the second attempt also fails. This new path is shown in the image below.



As an alternative to modifying a workflow to avoid loops, you can also use the Update Parameters configuration (see *Updating workflow parameters, page 623*). This configuration lets you update a parameter, including incrementally, which allows you to

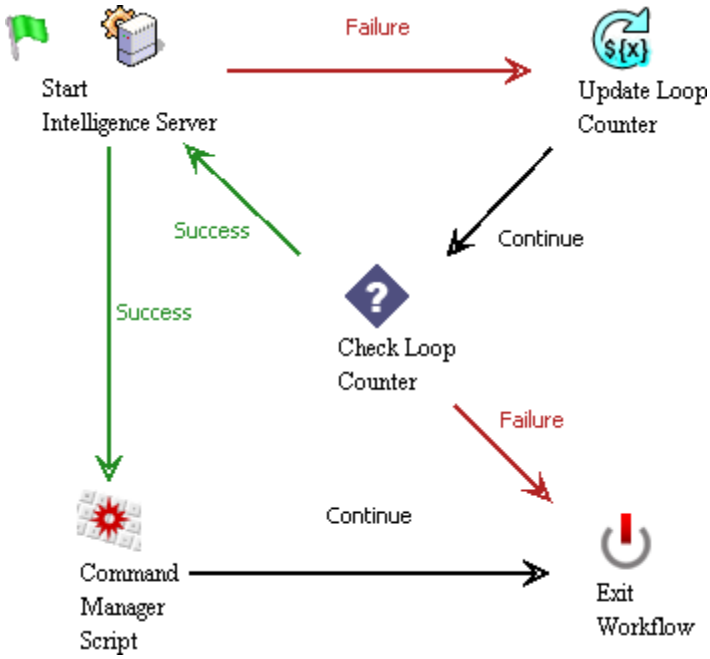
end a loop in a workflow after a specified number of attempts (see *Supporting loops in a workflow to attempt configurations multiple times, page 570* below).

Supporting loops in a workflow to attempt configurations multiple times

When deploying a workflow, it may be necessary to perform the same configuration multiple times. For example, if you attempt to start Intelligence Server but it does not start successfully, you can continue to attempt to start Intelligence Server until it starts successfully, or the workflow is ended. To support this type of a workflow, you can include a loop in your workflow.

Loops should commonly be avoided in workflows because they can cause a workflow to continue to perform the same actions repeatedly with no way to end the workflow. However, you can use decision processes and the Update Parameters process (see *Updating workflow parameters, page 623*) to support loops in workflows. By including the Update Parameters process in a workflow, you can keep track of how many times a loop in a workflow is repeated. After a certain amount of attempts, the loop can be exited even if the required configuration was not completed successfully.

For example, the workflow shown below uses a loop to attempt to start Intelligence Server, multiple times if necessary, before performing a Command Manager script that requires Intelligence Server to be operational.



With the workflow shown above, if Intelligence Server starts successfully the first time, the Command Manager script is executed next and the loop is not needed. However, if starting Intelligence Server is not successful, the first thing that occurs is that the Update Loop Counter configuration updates a parameter for the workflow. A parameter named Loop is included in the workflow with the initial value of zero, and the Update Loop Counter configuration updates this parameter with the following statement:

© 2016, MicroStrategy Inc.

```
${Loop} + 1
```

Using this statement, the Loop parameter is increased by one each time the Update Loop Counter configuration is executed. Once the Loop parameter has been increased, a decision process is used to check the value of the Loop parameter. If the Loop parameter is less than three, the configuration to start Intelligence Server is attempted again. This allows the configuration to start Intelligence Server to be attempted three times. If Intelligence Server still cannot start successfully, the loop is discontinued and the workflow is stopped.

To use split and merge executions in a workflow that uses logical loops, see *Processing multiple tasks simultaneously, page 562*.

Sample workflows: Templates

System Manager includes sample, template workflows that you can use to learn how to create workflows in System Manager and use as building blocks for your own workflows.

From the System Manager home page, you can access the template workflows in the Templates section. To choose from the full list of template workflows, click the **More Templates** folder.

Once the workflow is open in System Manager, you can select each process in the workflow to review the task that it performs for the workflow. You can also modify the properties of each process so that the workflow can be used to configure and administer your environment. For information on the properties available for each type of process available using System Manager, see *Defining processes, page 578*.

Overviews of the following templates are below:

- *Template: Configuring Intelligence Server, page 572*
- *Template: Configuring MicroStrategy Suite, page 572*
- *Template: Upgrading MicroStrategy Web, including customizations, page 573*
- *Template: Upgrading a metadata and executing an integrity test, page 573*
- *Template: Retrieving the status of Intelligence Server, page 574*
- *Template: Migrating objects between two projects and executing an integrity test, page 574*
- *Template: Including a cloud-based environment to increase Intelligence Server capacity, page 575*
- *Template: Restarting Intelligence Server, page 576*
- *Template: Updating projects with multiple update packages, page 576*
- *Template: Publishing Intelligent Cubes and workflow troubleshooting, page 577*
- *Template: Launching cloud-based environments in parallel, page 577*
- *Template: Creating and sharing update packages, page 578*

© 2016, MicroStrategy Inc.

Template: Configuring Intelligence Server

The template `01ConfigureIntelligenceServer.smw` can be used to configure Intelligence Server. The template includes the following tasks:

- Creates a new DSN to store a metadata.
- Configures Intelligence Server to connect to the new DSN.
- Creates a new project source, which allows access to the metadata.

Before using this template, be sure the following prerequisites are met:

- A database location used to store a MicroStrategy metadata. By default, the template creates a DSN for a Microsoft SQL Server database. You can swap in a process that matches the database type that you use to store your metadata. For a list of processes that can be used to create DSNs, see *Creating data source names, page 595*.
- Separate response files used to connect Intelligence Server to the new DSN and to create a new project source. These response files can be created using MicroStrategy Configuration Wizard, as described in the *Installation and Configuration Guide*.

Template: Configuring MicroStrategy Suite

The template `02ReportingSuiteSetup.smw` can be used to configure MicroStrategy Suite. The MicroStrategy Suite is a MicroStrategy offering that lets you evaluate MicroStrategy as a departmental solution. For detailed information to download, install, configure, and use the MicroStrategy Suite, see the *MicroStrategy Suite: Quick Start Guide*. This template includes the following tasks:

- Creates a new metadata.
- Configures Intelligence Server to connect to the new metadata.
- Creates a new project source, which allows access to the new metadata.
- Creates a new database instance for the MicroStrategy Suite.
- Creates a new project for the MicroStrategy Suite and connects it to the new database instance.
- Sends an email notification that describes the success of configuring the MicroStrategy Suite.

Before using this template, be sure the following prerequisites are met:

- Access to the MicroStrategy Suite software, which is available at http://www.microstrategy.com/freereportingsoftware/.
- Separate response files used to create a new metadata, connect Intelligence Server to the new DSN, and create a new project source. These response files can be created using MicroStrategy Configuration Wizard, as described in the *Installation and Configuration Guide*.
- Separate Command Manager scripts used to create a database instance, create a new project, and connect the new project to the new database instance. These scripts can

 © 2016, MicroStrategy Inc.

be created using Command Manager, as described in *Chapter 15, Automating Administrative Tasks with Command Manager*.

Template: Upgrading MicroStrategy Web, including customizations

The template `03UpgradeWebWithCustomizations.smw` can be used to upgrade your MicroStrategy Web environment. This upgrade workflow also supports including any customizations that you made to your MicroStrategy Web environment. This template includes the following tasks:

- Stops the web application server that hosts MicroStrategy Web.
- Creates a backup copy of MicroStrategy Web customization files.
- Creates a copy of the new web archive (.war) file to deploy the new version of MicroStrategy Web.
- Restarts the web application server, which extracts the contents of the .war file.
- Copies the MicroStrategy Web customization files into the newly deployed environment.
- Stops and then restarts the web application server, which deploys the new MicroStrategy Web environment, including any customizations.

Before using this template, be sure the following prerequisites are met:

- Access to any MicroStrategy Web customizations that are to be applied to the upgraded MicroStrategy Web environment. Review the MicroStrategy Software Development Library (MSDL) before upgrading MicroStrategy Web customizations for important upgrading best practices information.
- Access to the .war file for the version of MicroStrategy Web to upgrade to.
- A file used to start the web application server. By default, the template expects an Apache Tomcat web application server. You can swap in a file that starts your web application server.

Template: Upgrading a metadata and executing an integrity test

The template `04UpgradeMetadata.smw` can be used to upgrade a metadata and execute an integrity test after the upgrade is complete. This template includes the following tasks:

- Creates a backup copy of the metadata. An email is sent if a backup copy cannot be created.
- Upgrades the metadata. An email is sent if the upgrade is not completed successfully. As part of a successful upgrade, the backup file is compressed into a zip file, and the original backup file is deleted.
- Executes an Integrity Manager baseline test on the upgraded metadata.

Before using this template, be sure the following prerequisites are met:

© 2016, MicroStrategy Inc.

- Access to the metadata, and a SQL statement that can be used to create a copy of the metadata. By default, the template expects the metadata to be stored in a Microsoft SQL Server database. You can change the supplied SQL script to reflect the SQL syntax required for the database management system that you use to store your metadata.
- A response file used to upgrade the metadata. This response file can be created using MicroStrategy Configuration Wizard, as described in the *Installation and Configuration Guide*.
- A test file that defines how to perform the automated test of reports and documents for the metadata. This file can be created using Integrity Manager, as described in *Creating an integrity test, page 670*.

Template: Retrieving the status of Intelligence Server

The template `05IntelligenceServerAvailability.smw` can be used to retrieve the status of Intelligence Server and start Intelligence Server if it is not operational. This template includes the following tasks:

- Retrieves the status of Intelligence Server.
- Attempts to start Intelligence Server if it is not running.
- Sends an email notification that describes the success or failure of starting Intelligence Server.

Before using this template, be sure the following prerequisite is met:

- Access to an Intelligence Server.

Template: Migrating objects between two projects and executing an integrity test

The template `06ObjectMigration.smw` can be used to migrate objects between two projects and execute an integrity test after the object migration is complete. This template can be used to migrate a project from a testing environment to a production environment. This template includes the following tasks:

This template is not provided if System Manager is installed on a UNIX or Linux environment.

- Retrieves the status of Intelligence Server and attempts to start Intelligence Server if it is not operational. If Intelligence Server cannot be started, an email is sent and the workflow is ended.
- Merges two projects into a single project.
- Applies an update package to the merged project. An update package is a file containing a set of object definitions and conflict resolution rules.
- Restarts Intelligence Server and executes an Integrity Manager test on the merged project.

 © 2016, MicroStrategy Inc.

- Sends an email notification if any of the project migration steps fails.

Before using this template, be sure the following prerequisites are met:

- A file that defines how the duplicate projects are to be merged. This file is created using the Project Merge Wizard. For steps on how to create this configuration file, see *Merging projects with the Project Merge Wizard, page 267*.
- An update package file that defines how a project is to be duplicated. This file is created using MicroStrategy Object Manager. For steps on how to create this update package, see *Copying objects in a batch: Update packages, page 249*.
- A test file that defines how to perform the automated test of reports and documents for the project. This file can be created using Integrity Manager, as described in *Creating an integrity test, page 670*.

Template: Including a cloud-based environment to increase Intelligence Server capacity

The template `07AddIntelligenceServerCapacity.smw` can be used to include a cloud-based environment to increase Intelligence Server capacity. This template includes the following tasks:

- Launches an Amazon EC2 cloud-based environment, which can be used to increase Intelligence Server capacity.
- Ensures that MicroStrategy Listener is running, which is required to communicate with Intelligence Server.
- Attempts to start Intelligence Server.
- Searches through a response file used to create a project source. The Intelligence Server machine name is modified to match the machine name for the cloud-based environment.
- Creates a new project source to connect to the cloud-based environment.
- Searches through a Command Manager script file used to join the cloud-based environment to an Intelligence Server cluster. The Intelligence Server machine name is modified to match the machine name for the cloud-based environment.
- Joins the cloud-based environment to an Intelligence Server cluster.
- Sends an email notification that describes the success of adding the cloud-based environment to the Intelligence Server cluster.

Before using this template, be sure the following prerequisites are met:

- Access to an Amazon EC2 cloud-based environment, including all relevant support files and information. Refer to your third-party Amazon EC2 documentation for information on the requirements to support a cloud-based environment.
- A response file used to create a new project source. This response file can be created using MicroStrategy Configuration Wizard, as described in the *Installation and Configuration Guide*.

© 2016, MicroStrategy Inc.

- A Command Manager script file used to join the cloud-based environment to an Intelligence Server cluster. This script can be created using Command Manager, as described in *Creating and executing scripts, page 653*.

Template: Restarting Intelligence Server

The template `08IntelligenceServerRe-Start.smw` can be used to restart Intelligence Server and notify users of the scheduled restart. This template includes the following tasks:

- Sends an email to users as a warning that Intelligence Server is about to be restarted.
- Attempts to restart Intelligence Server and determines the success or failure of the restart.
- Sends an email to either the administrator or the broader user community, depending on whether the restart was successful.

Template: Updating projects with multiple update packages

The template `09MigrateMultiplePacakgesUsingLoop.smw` can be used to roll back a recent update package for multiple projects as well as apply a new update package. This template also serves as an example of successfully using loops in a System Manager workflow. This template includes the following tasks:

- Downloads update package files from an SFTP server.
- Creates a parameter that determines how many times the loop in the workflow has been completed. This parameter is used to choose the correct update packages and to exit the loop in the workflow at the proper time.
- Checks for all required update package files and undo package files.
- Sends an email to an administrator if some package files are not available.
- Modifies a Command Manager script to select a different undo package and update package for each loop through the workflow.
- Creates an undo package to roll back changes that were made to a project using an update package.
- Completes the undo package to roll back changes for the project, and then completes a new update package to update the objects for the project.
- Sends an email to an administrator verifying that the updates to the project were completed.
- Continues to loop through the workflow to do the same type of updates for other projects, or ends the workflow after updating four projects with these changes.

Before using this template, be sure the following prerequisites are met:

- Undo package files that define how to roll back the changes made by an update package for a project. This file is created using MicroStrategy Object Manager. For

 © 2016, MicroStrategy Inc.

steps on how to create this undo package, see *Rolling back changes: Undo packages, page 265*.

- Update package files that define how a project is to be updated. This file is created using MicroStrategy Object Manager. For steps on how to create this update package, see *Copying objects in a batch: Update packages, page 249*.
- Command Manager script files that are used to create and administer the undo package files. These script files can be created using Command Manager, as described in *Creating and executing scripts, page 653*.

Template: Publishing Intelligent Cubes and workflow troubleshooting

The template `10PublishCubesWithValidation.smw` can be used to publish Intelligent Cubes, and as an example of a workflow that uses the Decision process to troubleshoot the System Manager workflow. This template includes the following tasks:

- Employs an iterative retrieval process to retrieve information from a text file on the Intelligent Cubes to be published.
- Uses Command Manager script files to publish Intelligent Cubes.
- Uses multiple Decision processes to determine the success or failure of publishing the Intelligent Cubes.
- Sends emails about the success or failure of publishing the Intelligent Cubes.

Before using this template, be sure the following prerequisites are met:

- A text file that includes the information required to publish the Intelligent Cubes. Each line of the file must include two columns. The first column provides the Intelligent Cube name, and the second column provides the full path to the Command Manager script files used to publish the Intelligent Cube.
- Two Command Manager script files used to publish Intelligent Cubes. These script files can be created using Command Manager, as described in *Creating and executing scripts, page 653*.

Template: Launching cloud-based environments in parallel

The template `11ParallelExecutionOfWorkflows.smw` can be used to launch multiple cloud-based environments. It also is an example of using parallel execution in System Manager. This template includes the following tasks:

- Uses a split execution process to start two threads for the workflow to perform parallel processing.
- Launches two Amazon EC2 cloud-based environments in parallel, which can be used to increase Intelligence Server capacity.
- Checks to see if the cloud-based environments were launched successfully.
- Sends emails about the success or failure of launching the cloud-based environments.

Before using this template, be sure the following prerequisite is met:

- Access to an Amazon EC2 cloud-based environment, including all relevant support files and information. Refer to your third-party Amazon EC2 documentation for information on the requirements to support a cloud-based environment.

Template: Creating and sharing update packages

The template `12CreateSharePackage.smw` can be used to create a project update package and share that update package on an SFTP server. This template includes the following tasks:

- Retrieves the status of Intelligence Server.
- Attempts to start Intelligence Server if it is not running.
- Uses an `.xml` file to create an update package.
- Uploads the update package file to an SFTP server.
- Sends an email notification about the availability of the update package.

Before using this template, be sure the following prerequisites are met:

- Access to an Intelligence Server.
- Access to an SFTP server to store the update package.
- An `.xml` file that can be used to create an update package.
- A text file that includes a list of people to notify about the availability of the newly created update package.

Defining processes

The tasks that are completed as part of a System Manager workflow are determined by the processes that you include. System Manager provides a set of MicroStrategy and non-MicroStrategy processes to include in a workflow. These processes can be categorized as follows:

- *Configuring MicroStrategy components, page 579*
- *Managing projects, page 582*
- *Administering Intelligence Servers and other MicroStrategy services, page 586*
- *Automating administrative tasks, page 589*
- *Verifying reports and documents, page 592*
- *Creating data source names, page 595*
- *Completing a separate System Manager workflow, page 607*
- *Performing system processes, page 609*

 © 2016, MicroStrategy Inc.

- *Administering cloud-based environments, page 625*

System Manager workflows often require information about the result of a process to determine the next step to follow in the workflow. An exit code is provided when a process is completed that is part of a System Manager workflow. This exit code indicates whether the process was successful. For additional information on how to review the exit codes for a process, see *Determining process resolution using exit codes, page 637*.

Although all necessary configuration information can be provided for each process, some scenarios require that the details about the process be provided when the workflow is executed. Parameters provide the flexibility of including required configuration information when the workflow is executed. For information on how parameters can be used to provide configuration information for a process, see *Using parameters for processes, page 638*.

Configuring MicroStrategy components

After installing MicroStrategy, a few configurations need to be completed to set up a MicroStrategy environment. These tasks can be used to configure or upgrade MicroStrategy environments, as described below:

- *Creating metadata, History List, and statistics repository tables, page 579*
- *Configuring Intelligence Server, page 580*
- *Creating project sources, page 580*
- *Upgrading Intelligence Server components and migrating History List repositories, page 581*
- *Upgrading statistics repositories, page 581*
- *Migrating Narrowcast Server web delivery subscriptions to MicroStrategy Distribution Services, page 581*

Creating metadata, History List, and statistics repository tables

You can create metadata, History List, and statistics repositories as part of the process to configure a MicroStrategy environment. Repositories for your metadata, History List, and statistics tables are created in the data source specified by the DSNs that you connect to.

For background information on creating metadata, History List, and statistics repositories, see the *Installation and Configuration Guide*.

To perform these types of configurations, in System Manager, from the Connectors and processes pane, add the **Configuration Wizard** process to your workflow. The following information is required to create metadata, History List, and statistics repositories:

Metadata, History List, and statistics repositories can be part of the same process or included in their own separate processes in a System Manager workflow. Including

them as one process allows you to do all these configurations in a single process. However, including them in separate processes allows you to find and fix errors specific to each separate type of repository configuration and perform each configuration at different stages of the workflow.

- **Response File**: The MicroStrategy Configuration Wizard response file that defines how to create metadata, History List, and statistics repositories. Click the folder icon to browse to and select a response file. For information on how to create a Configuration Wizard response file, see the *Installation and Configuration Guide*.
- **Notes**: Information to describe this process as part of the workflow.

Configuring Intelligence Server

You can create, use, or delete server definitions that are used to provide a connection between Intelligence Server and your MicroStrategy metadata.

For background information on configuring Intelligence Server, see the *Installation and Configuration Guide*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Configuration Wizard** process to your workflow. The following information is required to create, use, or delete server definitions to configure Intelligence Server:

- **Response File**: The MicroStrategy Configuration Wizard response file that defines how to configure Intelligence Server. Click the folder icon to browse to and select a response file. For information on how to create a Configuration Wizard response file, see the *Installation and Configuration Guide*.
- **Notes**: Information to describe this process as part of the workflow.

Creating project sources

You can create project sources as part of your System Manager workflow. A project source contains the configuration information that each client system requires to access an existing project. It stores the location of the metadata repository and Intelligence Server that is used to run the project. A project source determines how Developer, MicroStrategy Web, and other client applications access the metadata.

For background information on creating project sources, see the *Installation and Configuration Guide*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Configuration Wizard** process to your workflow. The following information is required to create project sources:

- **Response File**: The MicroStrategy Configuration Wizard response file that defines how to create project sources. Click the folder icon to browse to and select a response file. For information on how to create a Configuration Wizard response file, see the *Installation and Configuration Guide*.
- **Notes**: Information to describe this process as part of the workflow.

 © 2016, MicroStrategy Inc.

Upgrading Intelligence Server components and migrating History List repositories

You can upgrade Intelligence Server components and migrate your History List from a file-based system to a database-based system. The Intelligence Server upgrade must be performed before any other upgrade or migration actions. For background information on upgrading MicroStrategy, see the *Upgrade Guide*.

To perform these types of configurations, in System Manager, from the Connectors and processes pane, add the **Configuration Wizard** process to your workflow. The following information is required to upgrade Intelligence Server components and migrate History List repositories:

- **Response File**: The MicroStrategy Configuration Wizard response file that defines how to upgrade Intelligence Server components or migrate your History List from a file-based system to a database-based system. Click the folder icon to browse to and select a response file. For information on how to create a Configuration Wizard response file, see the *Installation and Configuration Guide*.
- **Notes**: Information to describe this process as part of the workflow.

Upgrading statistics repositories

You can upgrade the statistics tables in your statistics repository to the new version of MicroStrategy. This statistics table upgrade ensures that your MicroStrategy Enterprise Manager environment can benefit from new features and enhancements in the most recent release of MicroStrategy.

You must perform an upgrade of your Intelligence Server components before any other upgrade or migration actions.

For background information on upgrading statistics repositories, see the *Upgrade Guide*. For information on Enterprise Manager, see the *MicroStrategy Operations Manager Guide*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Configuration Wizard** process to your workflow. The following information is required to upgrade statistics repositories:

- **Response File**: The MicroStrategy Configuration Wizard response file that defines how to upgrade the statistics tables in your statistics repository to the new version of MicroStrategy. Click the folder icon to browse to and select a response file. For information on how to create a Configuration Wizard response file, see *Installation and Configuration Guide*.
- **Notes**: Information to describe this process as part of the workflow.

Migrating Narrowcast Server web delivery subscriptions to MicroStrategy Distribution Services

You can migrate MicroStrategy web delivery subscriptions from a Narrowcast Server environment to Distribution Services. MicroStrategy web delivery subscriptions include

email, file, FTP, mobile, and print subscriptions created from MicroStrategy Web. These subscriptions are created when a user in MicroStrategy Web subscribes to a report or document.

Migrating these subscriptions from Narrowcast Server to Distribution Services allows the subscriptions to be centralized within Intelligence Server rather than a separate Narrowcast Server.

You must perform an upgrade of your Intelligence Server components before any other upgrade or migration actions.

For background information on migrating Narrowcast web delivery subscriptions to MicroStrategy Distribution Services, see the *Upgrade Guide*. For information on configuring and using Distribution Services, see *Configuring and administering Distribution Services, page 501*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Configuration Wizard** process to your workflow. The following information is required to migrate MicroStrategy web delivery subscriptions from a Narrowcast Server environment to Distribution Services:

- **Response File**: The MicroStrategy Configuration Wizard response file that defines how to migrate MicroStrategy web delivery subscriptions from a Narrowcast Server environment to Distribution Services. Click the folder icon to browse to and select a response file. For information on how to create a Configuration Wizard response file, see *Installation and Configuration Guide*.
- **Notes**: Information to describe this process as part of the workflow.

Managing projects

A MicroStrategy business intelligence application consists of many objects within projects. These objects are ultimately used to create reports and documents that display data to the end user. As in other software systems, these objects should be developed and tested before they can be used in a production system. Once in production, projects need to be managed to account for new requirements and previously unforeseen circumstances. This process is referred to as the project life cycle.

With System Manager, you can include these project management tasks in a workflow. This lets you create, manage, and update your projects silently, which can be done during off-peak hours and system down times. In performing project maintenance in this way, users of the MicroStrategy system are less affected by project maintenance.

For background information on managing your projects, see *Chapter 6, Managing Your Projects*.

System Manager supports the following project creation and maintenance tasks:

- *Merging duplicate projects to synchronize objects, page 583*
- *Duplicating projects, page 584*
- *Migrating projects to a new database platform, page 584*
- *Updating project objects, page 585*

 © 2016, MicroStrategy Inc.

- *Creating a package to update project objects, page 586*

Merging duplicate projects to synchronize objects

You can merge duplicate projects to synchronize many objects between duplicate projects as part of a System Manager workflow. This process migrates an entire project. All objects are copied to the destination project. Any objects that are present in the source project but not the destination project are created in the destination project.

For background information on merging duplicate projects, see *Merging projects to synchronize objects, page 266*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Project Merge (Windows Only)** process to your workflow. The following information is required to merge duplicate projects:

- **Project Merge XML File**: The file that defines how the duplicate projects are to be merged. This file is created using the Project Merge Wizard. For steps on how to create this configuration file, see *Merging projects with the Project Merge Wizard, page 267*.

For the password fields listed below, you can use the button to the right of the password fields to determine whether the password characters are shown or asterisks are displayed instead.

- **Source Project Source Password**: The password to access the source project source. The user name to access the source project source is provided in the configuration file created in Project Merge Wizard.
- **Destination Project Source Password**: The password to access the destination project source. The user name to access the destination project source is provided in the configuration file created in Project Merge Wizard.
- **Source Metadata Password**: The password to access the source metadata. The user name to access the source metadata is provided in the configuration file created in Project Merge Wizard.
- **Destination Metadata Password**: The password to access the destination metadata. The user name to access the destination metadata is provided in the configuration file created in Project Merge Wizard.
- **Update the metadata if the metadata of the destination project is older than the source project**: Forces a metadata update of the destination metadata if it is older than the source metadata and this check box is selected. The merge is not executed unless the destination metadata is the same version as or more recent than the source metadata.
- **Update the schema of the destination project at the end**: If this check box is selected, the system updates the schema of the destination project after the merge is completed. This update is required when you make any changes to schema objects such as facts, attributes, or hierarchies.

© 2016, MicroStrategy Inc.

Do not use this option if the configuration file contains an instruction to update the schema.

- **Forcefully take over locks if any of the sessions are locked**: If this check box is selected, the system takes ownership of any metadata locks that exist on the source or destination projects. If this check box is cleared and sessions are locked, the project merge cannot be completed.
- **Notes**: Information to describe this process as part of the workflow.

Duplicating projects

You can duplicate projects as part of a System Manager workflow. If you want to copy objects between two projects, MicroStrategy recommends that the projects have related schemas. This means that one must have originally been a duplicate of the other, or both must have been duplicates of a third project.

For background information on duplicating projects, see *Duplicating a project, page 225*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Project Duplication (Windows Only)** process to your workflow. The following information is required to duplicate projects:

- **XML Configuration File**: The file that defines how a project is to be duplicated. This file is created using the Project Duplication Wizard. For steps on how to create this configuration file, see *The Project Duplication Wizard, page 227*.
- **Base Project Password**: The password for the source project's project source. You can use the button to the right of this password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Target Project Password**: The password for the destination project's project source. You can use the button to the right of this password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Update Target Metadata**: If this check box is selected, the system forces a metadata update of the destination metadata if it is older than the source metadata. The duplication is not executed unless the destination metadata is the same version as or more recent than the source metadata.
- **Overwrite the project name specified in the configuration file**: The new name to use for the destination project. Select the check box and type a new name to replace the name specified in the XML settings file.
- **Notes**: Information to describe this process as part of the workflow.

Migrating projects to a new database platform

You can migrate, as part of a System Manager workflow, a MicroStrategy project that is stored in a Microsoft Access database to a new database platform. This can be useful for creating and testing a new project before it is placed in the production database.

 © 2016, MicroStrategy Inc.

To migrate a project to a new database platform, you must have the Bypass All Object Security Access Checks privilege for that project.

For background information on migrating projects to a new database platform, see *Migrating a project to a new database platform, page 228*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Project Mover (Windows Only)** process to your workflow. The following information is required to migrate a MicroStrategy project stored in a Microsoft Access database to a new database platform:

- **Response File**: The MicroStrategy Project Mover response file that defines how to migrate a MicroStrategy project stored in a Microsoft Access database to a new database platform as part of a System Manager workflow. Click the folder icon to browse to and select a response file. For information on how to create a Project Mover response file, see *Migrating a project automatically, page 229*.
- **Notes**: Information to describe this process as part of the workflow.

Updating project objects

You can use an update package as part of a System Manager workflow. An update package is a file containing a set of object definitions and conflict resolution rules. It allows you to save the objects you want to copy in an update package and import that package into destination projects later.

For background information on updating projects using update packages, see *Copying objects in a batch: Update packages, page 249*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Import Package** process to your workflow. The following information is required to update a project using an upgrade package:

- **Project Source Name**: The name of the project source that contains the project to update objects in using the update package.
- **Login**: The name of a valid user to log in to the project source.
- **Password**: The password for the user name that you provided to log in to the project source. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Package file**: The update package file that defines how a project is to be duplicated. This file is created using MicroStrategy Object Manager. For steps to create this update package, see *Copying objects in a batch: Update packages, page 249*.

  If you are importing a package that is stored on a machine other than the Intelligence Server machine, ensure that the package can be accessed by the Intelligence Server machine.

- **Destination Project Name**: Determines whether the update package is a project update package or a configuration update package:

    - If the update package is a project update package, select this check box and type the name of the project to update objects in using the update package.
    - If the update package is a configuration update package, clear this check box.

- **Use logging**: If this check box is selected, the system logs the update package process. Click the folder icon to browse to and select the file to save the update package results to. If this check box is cleared, no log is created.
- **Forcefully acquire locks**: If this check box is selected, the system takes ownership of any locks that exist. If this check box is cleared and sessions are locked, the update package cannot be completed.
- **Notes**: Information to describe this process as part of the workflow.

Creating a package to update project objects

You can create an update package as part of a System Manager workflow. An update package is a file containing a set of object definitions and conflict resolution rules. It allows you to save the objects you want to copy in an update package, and import that package into any number of destination projects at a later date.

For background information on creating update packages, see *Creating an update package, page 251*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Create Package** process to your workflow. The following information is required to create an upgrade package:

- **Package XML File**: The `.xml` file that contains the definition to create a package file. You can use Object Manager to create this `.xml` file, as described in *Creating an update package from the command line, page 255*.
- **Source Project Source Password**: The password for the user account you used to create the package `.xml` file. This authentication information is used to log in to the project source. You can use the button to the right of the password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Source Metadata Password**: The password for the user account you used to create the package `.xml` file. This authentication information is used to log in to the project metadata. You can use the button to the right of the password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Notes**: Information to describe this process as part of the workflow.

Administering Intelligence Servers and other MicroStrategy services

Intelligence Server and other MicroStrategy services are required to be operational to complete certain processes that can be a part of a System Manager workflow. To support

 © 2016, MicroStrategy Inc.

this requirement, you can include the administration of Intelligence Server and other MicroStrategy services as part of a workflow.

The types of MicroStrategy services administration that you can perform as part of a workflow include:

- *Starting, stopping, or restarting MicroStrategy services, page 587*
- *Determining the status of MicroStrategy services, page 588*

Starting, stopping, or restarting MicroStrategy services

You can start, stop, or restart MicroStrategy services as part of a System Manager workflow. This can help to ensure Intelligence Server is operational, which is required to perform various processes. You can also stop Intelligence Server to make system-wide updates, and then restart Intelligence Server once all the updates are made.

To perform these types of configuration, in System Manager, from the Connectors and processes pane, add the **Manage MicroStrategy Service** process to your workflow. The following information is required to start, stop, or restart a MicroStrategy service:

- **Action**: Determines whether to start, stop, or restart the MicroStrategy service. Select the required action from this drop-down list.
- You can determine which machine to administer its services for using one of the following options:
    - **Local machine**: This option performs the start, stop, or restart action for the service of the machine used to deploy the workflow.
    - **Remote machine**: This option lets you specify the machine that hosts the service to perform the start, stop, or restart action for. You must provide the information listed below:
        - **Machine Name**: The name of the machine that hosts the service.
        - **Login**: The name of a valid user to administer the service.
        - **Password**: The password for the user name that you provided to administer the service. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Service Type**: Determines the service to start, stop, or restart. From this drop-down list, you can select one of the following MicroStrategy services:
    - **MicroStrategy Intelligence Server**: The main service for your MicroStrategy reporting environment. It provides the authentication, clustering, governing, and other administrative management requirements for your MicroStrategy reporting environment.
    - **MicroStrategy Listener**: Also known as Test Listener. A ping utility that allows you to check the availability of an Intelligence Server on your network, whether a DSN can connect to a database, and whether a project source name can connect to a project source. From any machine that has the Test Listener

installed and operational, you can get information about other MicroStrategy services available on the network without having to actually go to each machine.

  - **MicroStrategy Enterprise Manager Data Loader**: The service for Enterprise Manager that retrieves data for the projects for which statistics are being logged. This data is then loaded into the Enterprise Manager lookup tables for further Enterprise Manager reporting and analysis.

  - **MicroStrategy Distribution Manager**: The service for Narrowcast Server that distributes subscription processing across available Execution Engines.

  - **MicroStrategy Execution Engine**: The service for Narrowcast Server that gathers content, formats the content, and delivers the content to the devices for a subscription.

- **Notes**: Information to describe this process as part of the workflow.

Determining the status of MicroStrategy services

You can retrieve the status of a MicroStrategy service as part of a System Manager workflow. This can help to ensure that required MicroStrategy services are operational, which can be required to perform various processes.

To retrieve this information, in System Manager, from the Connectors and processes pane, add the **Get Service Status** process to your workflow. The following information is required to retrieve the status of a MicroStrategy service:

- You can determine the machine for which to retrieve the service status by using one of the following options:

  - **Local machine**: Retrieves the status for the service of the machine used to deploy the workflow.

  - **Remote machine**: Lets you specify the machine that hosts the service to retrieve the status for the service. If you select this option, you must type the name of the machine that hosts the service.

- **Service type**: Determines the service to retrieve the status for. From this drop-down list, you can select one of the following MicroStrategy services:

  - **MicroStrategy Intelligence Server**: The main service for your MicroStrategy reporting environment. It provides the authentication, clustering, governing, and other administrative management requirements for your MicroStrategy reporting environment.

  - **MicroStrategy Listener**: A ping utility that allows you to check the availability of an Intelligence Server on your network, whether a DSN can connect to a database, and whether a project source name can connect to a project source. From any machine that has the Test Listener installed and operational, you can get information about other MicroStrategy services available on the network without having to actually go to each machine.

  - **MicroStrategy Enterprise Manager Data Loader**: The service for Enterprise Manager that retrieves data for the projects for which statistics are being logged.

 © 2016, MicroStrategy Inc.

This data is then loaded into the Enterprise Manager lookup tables for further Enterprise Manager reporting and analysis.

  - **MicroStrategy Distribution Manager**: The service for Narrowcast Server that distributes subscription processing across available Execution Engines.
  - **MicroStrategy Execution Engine**: The service for Narrowcast Server that gathers content, formats the content, and delivers the content to the devices for a subscription.

- **Notes**: Information to describe this process as part of the workflow.

Automating administrative tasks

You can perform various administrative and application development tasks by using text commands that can be saved as scripts or entered as commands to be completed as part of a System Manager workflow. These scripts and commands are created using Command Manager.

For an introduction to Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*.

The types of processes that you can automate using Command Manager statements include:

- *Managing configurations for project sources, page 589*
- *Managing configurations for project sources using Command Manager Runtime statements, page 591*
- *Managing configurations for Narrowcast Server metadatas, page 591*

Managing configurations for project sources

You can use text commands as part of a System Manager workflow as a script or entered directly as statements, to add, delete, or update large numbers of users and user groups, as well as manage various configuration settings for project sources.

For an introduction to Command Manager, see *Chapter 15, Automating Administrative Tasks with Command Manager*.

To perform this configuration, in System Manager, from the Connectors and Processes pane, add the **Intelligence Server** process to your workflow. The following information is required to execute a Command Manager script or statements:

- **Connection Information**: Determines whether to use a connection-less session or to connect directly to a project source:
  - **Connection-less Session**: Defines the script execution as a connection-less session, which means a connection is not immediately made to a project source. A connection is required to perform any tasks included in the commands. You can use this option when the Command Manager statements include the required connection information.

© 2016, MicroStrategy Inc.

    - **Connect To A Project Source**: Defines the project source to connect to for the statement execution. Provide the following information:
        - **Project Source**: The name of the project source to connect to.
        - **Login**: The name of a valid user to connect to the project source.
        - **Password**: The password for the user name that you provided to connect to the project source. Use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.

MicroStrategy does not recommend using quotation marks in your passwords. If you are running MicroStrategy in Windows and your password contains one or more quotation marks ("), you must replace them with two quotation marks ("") and enclose the entire password in quotes. For example, if your password is 1"2"3'4'5, you must enter the password as "1""2""3'4'5".

- **Execution**: Choose whether the Command Manager statements to run are in a script file or the ones that you enter here:
    - **Script File (.scp)**: Browse to and select the Command Manager script file that defines all the tasks to be completed.
    - **Execute script statements**: Type in the Command Manager statement or statements to be completed.
- **Export Results To an XML File**: If selected, the system logs the execution results, error messages, and status messages to a single XML file. Click the folder icon to browse to and select an XML file.
- **Display Output On The Console**: If selected and the script is not encrypted, the system displays the results on the command line used to execute the script or statements. For information on encrypting a script, click Help from within Command Manager.
- **Stop Script Execution On Error**: If execution causes a critical error and this check box is selected, the system terminates the execution. Clear this check box to allow the execution to continue even if critical errors are encountered. For information on what constitutes a critical error, with the Command Manager interface open, press F1 to open and review the *Command Manager Help*.
- **Suppress Hidden Object(s) In The Results**: If this check box is selected, the system omits hidden objects in the execution results. Hidden objects are MicroStrategy metadata objects whose `HIDDEN` property is set as true.
- **Logging Information**: Defines how the results of running the Command Manager script or statements are logged. Select one of the following options:
    - **Log Output To Default Location**: Logs all results to the default folder.
    - **Log Output To Specified File**: Logs all results to the log file specified. You can browse to and select a log file.

 © 2016, MicroStrategy Inc.

  - **Split Output Into Three Defaults (Results, Failure, and Success)**: Logs all results to three separate log files. The default log files are `CmdMgrResults.log`, `CmdMgrFail.log`, and `CmdMgrSuccess.log`, respectively.
  - **Split Output Into Three Specified Files**: Logs all results of execution to three separate log files that you choose:
    - **Results File**: Includes any information provided by successful LIST statements that were executed.
    - **Failure File**: Includes a list of statements that were not executed successfully.
    - **Success File**: Includes a list of statements that were executed successfully.
  - **Include Instructions In The Log File(s)**: If selected, the system includes the statements in the log file or files.
  - **Include File Log Header**: If selected, the system includes a header at the beginning of each log file that contains information such as the version of Command Manager used.
  - **Include Error Codes in the Log File(s)**: If selected, the system includes any error codes returned during the workflow in the log file or files.
- **Notes**: Information to describe this process as part of the workflow.

Managing configurations for project sources using Command Manager Runtime statements

Developers of OEM applications that use embedded MicroStrategy projects may need flexibility in configuring their environment. Command Manager Runtime is a slimmed-down version of the Command Manager command-line executable for use with these OEM applications.

Command Manager Runtime uses a subset of the commands that are available in the full version of Command Manager. Command Manager Runtime statements can be included in a System Manager workflow as a script or as statements entered in the workflow.

To perform this configuration, in System Manager, from the Connectors and Processes pane, add the **Intelligence Server (Runtime)** process to your workflow. The information required to execute a Command Manager script or statements is the same information required for a standard Command Manager script or statements, which is described in *Managing configurations for project sources, page 589*. If you try to execute statements that are not available in Command Manager Runtime as part of a System Manager workflow, the script or statements fail with an exit code of 12.

For information on Command Manager Runtime, refer to the *Supplemental Reference for System Administration*.

Managing configurations for Narrowcast Server metadatas

MicroStrategy Command Manager lets you manage various configuration settings within the MicroStrategy platform, for Narrowcast Servers.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Narrowcast Server (Windows Only)** process to your workflow. The information required to execute a Command Manager script or statements used to manage Narrowcast Servers includes the same information required for Command Manager script or statements used to manage project sources, which is described in *Launching cloud-based environments, page 628*. In addition to this required information, Command Manager scripts or statements used to manage Narrowcast Servers also require the following information:

- **DSN**: The data source name that points to the database that stores the Narrowcast Server repository. If the DSN requires specific permissions, select the **Authentication for DSN** check box to provide a valid user name and password.
- **Database**: The database that stores the Narrowcast Server repository. Type the name of the database that resides in the DSN you specified in the DSN field. The DSN field is part of the options described in *Launching cloud-based environments, page 628*.
- **System Prefix**: The database prefix used to identify the Narrowcast Server repository.

Verifying reports and documents

You can run automated tests to determine how specific changes in a project environment, such as the regular maintenance changes to metadata objects or hardware and software upgrades, affect the reports and documents in that project as part of a System Manager workflow. These types of test can ensure that reports and documents are working as intended, as well as determining the performance of any new or updated MicroStrategy deployments.

For background information on running automated tests of reports and documents using MicroStrategy Integrity Manager, see *Chapter 16, Verifying Reports and Documents with Integrity Manager*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Integrity Manager** process to your workflow. The following information is required to run an automated test of reports and documents:

- **MTC Configuration File**: The test file that defines how to perform the automated test of reports and documents. This file is created using Integrity Manager. For steps on how to create this test file, see *Creating an integrity test, page 670*.
- **Base Project Password**: The password for the user specified in the test file to log in to the base project. This is not required for a baseline-versus-project or baseline-versus-baseline integrity test. You can use the button to the right of this password field to determine whether the password characters are shown or asterisks are displayed instead. Refer to *Specifying passwords for multiple user accounts and special characters, page 594* below for information on providing multiple passwords or passwords that use special characters for an Integrity Manager test.
- **Target Project Password**: The password for the user specified in the test file to log in to the destination project. This is not required for a single-project or baseline-versus-baseline integrity test. You can use the button to the right of this password field to determine whether the password characters are shown or asterisks are displayed instead. Refer to *Specifying passwords for multiple user accounts and special*

 © 2016, MicroStrategy Inc.

characters, page 594 below for information on providing multiple passwords or passwords that use special characters for an Integrity Manager test.

- You can use the following parameters to provide alternative test information and details when running an Integrity Manager test as part of a workflow. All parameters are optional, and if you clear the check box for a parameter listed below, any required information for that parameter is provided by the Integrity Manager test file instead:
  - **Output Directory**: The directory for any results. Click the folder icon to browse to and select an output directory.
  - **Log File**: Click the folder icon to browse to and select a log file directory.
  - **Base Baseline File**: Click the folder icon to browse to and select a baseline file for the base project.
  - **Target Baseline File**: Click the folder icon to browse to and select a baseline file for the target project.
  - **Base Server Name**: The name of the machine that is running the Intelligence Server that hosts the base project for the test.
  - **Base Server Port**: The port that Intelligence Server is using. The default port is 34952.
  - **Target Server Name**: The name of the machine that is running the Intelligence Server that hosts the target project for the test.
  - **Target Server Port**: The port that Intelligence Server is using. The default port is 34952.
  - **Base Project Name**: The name of the base project for the test.
  - **Login(s) for Base Project**: The login accounts required to run any reports or documents in the base project for the test. For multiple logins, enclose all logins in double quotes (`""`) and separate each login with a comma (`,`).
  - **Target Project Name**: The name of the target project for the test.
  - **Login(s) for Target Project**: The login accounts required to run any reports or documents in the base project for the test. For multiple logins, enclose all logins in double quotes (`""`) and separate each login with a comma (`,`).
  - **Test Folder GUID**: The GUID of the test folder. If this option is used, the reports and documents specified in the Integrity Manager test file are ignored. Instead, Integrity Manager executes all reports and documents in the specified folder.

    This option can only be used with a single-project integrity test or a project-versus-project integrity test.
  - **Load Balancing for Base Server**: Determines whether to use load balancing for the base server. If this option is used, it overrides the setting in the Integrity Manager test file.

    - **Load Balancing for Target Server**: Determines whether to use load balancing for the target server. If this option is used, it overrides the setting in the Integrity Manager test file.

- **Notes**: Information to describe this process as part of the workflow.

Specifying passwords for multiple user accounts and special characters

An Integrity Manager test can include multiple user accounts as part of the test, as well as user accounts that include special characters in their passwords.

To use multiple user accounts for testing, the passwords associated with each user account must also be provided. If your Integrity Manager test includes multiple user accounts, use the following rules to provide any required passwords for the base project and target project:

- You must include a password for each user account defined in the Integrity Manager test configuration file. However, if all user accounts use a blank password, you can leave the base project and target project password fields blank to indicate that a blank password is used for each user account.
- Enclose the full list of passwords in double quotes (").
- Separate each password using a comma (,).
- The passwords must be listed in the order that user accounts are defined in the Integrity Manager test. Use Integrity Manager to review the test file as required to determine the proper order.
- If a subset of user accounts use blank passwords, use a space to indicate a blank password. For example, if the second user account included in an Integrity Manager test has a blank password, you can define the password list as:

  ```
  "password1, ,password3"
  ```

An Integrity Manager test can include user accounts that include special characters in their passwords. Use the following rules to denote special characters in passwords for the base project and target project:

- If a password includes a single quote (') or comma (,), you must enclose the entire password in single quotes. For example, for the password `sec,ret`, you must type this password as `'sec,ret'`.
- To denote a single quote (') in a password, use two single quotes. For example, for the password `sec'ret`, you must type this password as `'sec''ret'`.
- To denote a double quote (") in a password, type `"`. For example, for the password `sec"ret`, you must type this password as `sec"ret`.
- To denote an ampersand (&) in a password, type `&`. For example, for the password `sec&ret`, you must type this password as `sec&ret`.

 © 2016, MicroStrategy Inc.

Creating data source names

Establishing communication between MicroStrategy and your databases or other data sources is an essential step in configuring MicroStrategy products for reporting and analysis of your data. A *data source name* (DSN) allows MicroStrategy to connect and communicate to your data sources. For background information on creating and supporting DSNs, see the *Installation and Configuration Guide*.

System Manager allows you to create DSNs for the following types of databases:

- *DB2 UDB, page 595*
- *UDB iSeries/DB2 for i, page 596*
- *DB2 z/OS , page 597*
- *Greenplum, page 598*
- *Hive, page 598*
- *Informix, page 599*
- *Informix XPS , page 600*
- *Microsoft SQL Server, page 600*
- *MySQL, page 603*
- *Oracle, page 603*
- *PostgreSQL, page 605*
- *Salesforce, page 605*
- *Sybase ASE, page 606*

Creating a DSN using System Manager can be successful or unsuccessful for various reasons, which are denoted using exit codes. For information on determining the possible exit codes of creating a DSN using System Manager, see *Determining process resolution using exit codes, page 637*.

DB2 UDB

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **DB2 UDB** process to your workflow. The following information is required to create a DSN for DB2 UDB when running against DB2:

- **Data Source Name**: A name to identify the DB2 UDB data source configuration in MicroStrategy. For example, Finance or DB2-Serv1 can serve to identify the connection.
- **IP Address**: The IP address or name of the machine that runs the DB2 UDB server.
- **TCP Port**: The DB2 UDB server listener's port number. In most cases, the default port number is **50000**, but you should check with your database administrator for the correct number.

© 2016, MicroStrategy Inc.

- **Database Name**: The name of the database to connect to by default, which is assigned by the database administrator.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created, and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
  - **Username**: The name of a valid user for the database.
  - **Password**: The password for the user name that you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Notes**: Information to describe this process as part of the workflow.

UDB iSeries/DB2 for i

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **DB2 UDB iSeries** process to your workflow. The following information is required to create a DSN for UDB iSeries/DB2 for i:

- **Data Source Name**: A name to identify the DB2 for i data source configuration in MicroStrategy. For example, Finance or DB2fori-1 can serve to identify the connection.
- **IP Address**: The IP Address of the machine where the catalog tables are stored. This can be either a numeric address, such as `123.456.789.98`, or a host name. If you use a host name, it must be in the `HOSTS` file of the machine or a DNS server.
- **Collection**: The name that identifies a logical group of database objects.
- **Location**: The DB2 location name, which is defined during the local DB2 installation.
- **Isolation Level**: The method by which locks are acquired and released by the system.
- **Package Owner**: The package's AuthID if you want to specify a fixed user to create and modify the packages on the database. The AuthID must have authority to execute all the SQL in the package.
- **TCP Port**: The DB2 DRDA listener process's port number on the server host machine provided by your database administrator. The default port number is usually **446**.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created, and the DSN is not updated.
- **Test Connection**: Tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is

 © 2016, MicroStrategy Inc.

performed. If this check box is selected, you must provide the following connection information:

  - **Username**: The name of a valid user for the database.
  - **Password**: The password for the user name that you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.

- **Notes**: Information to describe this process as part of the workflow.

DB2 z/OS

To perform this configuration, in System Manager, from the **Connectors and processes** pane, add the **DB2 z/OS** process to your workflow. The following information is required to create a DSN for DB2 z/OS:

- **Data Source Name**: A name to identify the DB2 z/OS data source configuration in MicroStrategy. For example, Finance or DB2UDBz/OS-1 can serve to identify the connection.
- **IP Address**: The IP Address of the machine where the catalog tables are stored. This can be either a numeric address such as `123.456.789.98`, or a host name. If you use a host name, it must be in the `HOSTS` file of the machine or a DNS server.
- **Collection**: The name that identifies a logical group of database objects, which is also the current schema. On DB2 z/OS, the user ID should be used as the Collection.
- **Location**: The DB2 z/OS location name, which is defined during the local DB2 z/OS installation. To determine the DB2 location, you can run the command `DISPLAY DDF`.
- **Package Collection**: The collection or location name where bind packages are created and stored for searching purposes.
- **Package Owner**: The package's AuthID if you want to specify a fixed user to create and modify the packages on the database. The AuthID must have authority to execute all the SQL in the package.
- **TCP Port**: The DB2 DRDA listener process's port number on the server host machine provided by your database administrator. The default port number is usually **446**.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
  - **Username**: The name of a valid user for the database.

© 2016, MicroStrategy Inc.

    - **Password**: The password for the user name that you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.

- **Notes**: Information to describe this process as part of the workflow.

Greenplum

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Greenplum** process to your workflow. The following information is required to create a DSN for Greenplum:

- **Data Source Name**: A name to identify the Greenplum data source configuration in MicroStrategy. For example, `Finance` or `Greenplum-1` can serve to identify the connection.
- **Host Name**: The name or IP address of the machine on which the Greenplum data source resides. The system administrator or database administrator assigns the host name.
- **Port Number**: The port number for the connection. The default port number for Greenplum is usually **5432**. Check with your database administrator for the correct number.
- **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
    - **Username**: The name of a valid user for the database.
    - **Password**: The password for the user name that you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Notes**: Information to describe this process as part of the workflow.

Hive

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Hive** process to your workflow. The following information is required to create a DSN for Apache Hive:

- **Data Source Name**: A name to identify the Apache Hive data source configuration in MicroStrategy. For example, `Finance` or `ApacheHive-1` can serve to identify the connection.

 © 2016, MicroStrategy Inc.

- **Host Name**: The name or IP address of the machine on which the Apache Hive data source resides. The system administrator or database administrator assigns the host name.
- **Port Number**: The port number for the connection. The default port number for Apache Hive is usually **10000**. Check with your database administrator for the correct number.
- **Database Name**: The name of the database to connect to by default. If no database name is provided, the default database is used for the connection. The database administrator assigns the database name.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed.
- **Notes**: Information to describe this process as part of the workflow.

Informix

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Informix** process to your workflow. The following information is required to create a DSN for Informix Wire Protocol:

- **Data Source Name**: A name to identify the Informix data source configuration in MicroStrategy. For example, Finance or Informix-1 can serve to identify the connection.
- **Server Name**: The client connection string designating the server and database to be accessed.
- **Host Name**: The name of the machine on which the Informix server resides. The system administrator or database administrator assigns the host name.
- **Port Number**: The Informix server listener's port number. The default port number for Informix is commonly **1526**.
- **Database Name**: The name of the database to connect to by default, which is assigned by the database administrator.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
    - **Username**: The name of a valid user for the database.

    - **Password**: The password for the user name that you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.

- **Notes**: Information to describe this process as part of the workflow.

Informix XPS

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Informix XPS (Windows Only)** process to your workflow. The following information is required to create a DSN for Informix XPS:

- **Data Source Name**: A name to identify the Informix data source configuration in MicroStrategy. For example, Finance or Informix-1 can serve to identify the connection.
- **Database**: The name of the database to connect to by default, which is assigned by the database administrator.
- **Server Name**: The client connection string designating the server and database to be accessed.
- **Host Name**: The name of the machine on which the Informix server resides. The system administrator or database administrator assigns the host name.
- **Service Name**: The service name, as it exists on the host machine. The system administrator assigns the service name.
- **Protocol Type**: The protocol used to communicate with the server. Select the appropriate protocol from this drop-down list.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
    - **Username**: The name of a valid user for the database.
    - **Password**: The password for the user name that you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Notes**: Information to describe this process as part of the workflow.

Microsoft SQL Server

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Microsoft SQL Server** process to your workflow. The following information is required to create a DSN for Microsoft SQL Server:

 © 2016, MicroStrategy Inc.

- **Data Source Name**: A name to identify the Microsoft SQL Server data source configuration in MicroStrategy. For example, Personnel or SQLServer-1 can serve to identify the connection.
- **Windows**: Select this option if you are configuring the Microsoft SQL Server driver on Windows:
  - **Server Name**: The name of a SQL Server on your network, in the format `ServerName_or_IPAddress,PortNumber`. For example, if your network supports named servers, you can specify an address such as `SQLServer-1,1433`. You can also specify the IP address such as `123.45.678.998,1433`.

    Additionally, if you use named instances to distinguish SQL Server databases, you can include the named instance along with either the server name or IP address using the format `ServerName\NamedInstance` or `IPAddress\NamedInstance`. The following are examples of providing the server name for your SQL Server database:

    — `SQLServer-1\Instance1,1433`

    — `123.45.678.998\Instance1,1433`
  - **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.
  - **Use Windows NT authentication for login**: Select this check box to use Windows NT authentication to pass a user's credentials on the Windows machine to execute against a SQL Server database.

    If you use Windows NT authentication with SQL Server, you must enter the Windows NT account user name and password in Service Manager. For background information on Service Manager, see *Service Manager, page 15*.
- **UNIX**: Select this option if you are configuring the MicroStrategy-branded version of the Microsoft SQL Server driver for use on UNIX and Linux:
  - **Server Name**: The name of a SQL Server on your network. For example, if your network supports named servers, you can specify an address such as `SQLServer-1`. You can also specify the IP address such as `123.45.678.998`. Contact your system administrator for the server name or IP address.

    Additionally, if you use named instances to distinguish SQL Server databases, you can include the named instance along with either the server name or IP address using the format `ServerName\NamedInstance` or `IPAddress\NamedInstance`. The following are examples of providing the server name for your SQL Server database:

    — `SQLServer-1\Instance1`

    — `123.45.678.998\Instance1`
  - **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.

© 2016, MicroStrategy Inc.

    - **Port Number**: The port number for the connection. The default port number for SQL Server is usually **1433**. Check with your database administrator for the correct number.
    - **Enable SQL Database (Azure) support**: Defines whether the DSN is created to support SQL Azure. Select this check box if the DSN is used to access a SQL Azure data source.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
    - **Username**: The name of a valid user for the database.
    - **Password**: The password for the user name that you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Notes**: Information to describe this process as part of the workflow.

Microsoft Access

The MicroStrategy ODBC Driver for SequeLink allows you to access Microsoft Access databases stored on a Windows machine from an Intelligence Server hosted on a UNIX or Linux machine.

Steps on how to perform the necessary configurations on the various machines to support this type of configuration are provided in the *Installation and Configuration Guide*.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Microsoft Access (Windows Only)** process to your workflow. The following information is required to create a DSN for Microsoft Access:

- **Data Source Name**: A name to identify the Microsoft SQL Server data source configuration in MicroStrategy. For example, Personnel or MicrosoftAccess-1 can serve to identify the connection.
- **Database**: The name of the database to connect to by default. Click the folder icon to browse to and select a Microsoft Access database.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: Tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed.

 © 2016, MicroStrategy Inc.

- **Notes**: Information to describe this process as part of the workflow.

MySQL

The MicroStrategy ODBC Driver for MySQL Wire Protocol is for use with MySQL Enterprise on Windows and Linux.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **MySQL** process to your workflow. The following information is required to create a DSN for MySQL:

- **Data Source Name**: A name to identify the MySQL data source configuration in MicroStrategy. For example, Finance or MySQL-1 can serve to identify the connection.
- **Host Name**: The name or IP address of the machine on which MySQL Enterprise 5.0 resides. The system administrator or database administrator assigns the host name.
- **Port Number**: The port number for the connection. The default port number for MySQL is usually **3306**. Check with your database administrator for the correct number.
- **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.
- **Default User ID**: The name of a valid user for MySQL Enterprise.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
  - **User Name**: The name of a valid user for the database.
  - **Password**: The password for the default user name that you provided. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Notes**: Information to describe this process as part of the workflow.

Oracle

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Oracle** process to your workflow. The following information is required to create a DSN for Oracle Wire Protocol:

- **Data Source Name**: A name to identify the Oracle data source configuration in MicroStrategy. For example, Finance or Oracle-1 can serve to identify the

connection. A DSN is required for any Oracle Wire Protocol connection. Depending on whether you want to use a standard connection or a TNSNames connection, refer to one of the following lists of options below:

- **Standard Connection**: A standard connection is configured through Oracle Wire Protocol with the following connection parameters:
  - **Host Name**: The name of the Oracle server to be accessed. This can be a server name such as Oracle-1 or an IP address such as `123.456.789.98`.
  - **Port Number**: The Oracle listener port number provided by your database administrator. The default port number is usually **1521**.
  - One of the following parameters; which one you choose is up to your personal preference:

SID: The Oracle System Identifier for the instance of Oracle running on the server. The default SID is usually **ORCL**.

Service Name: The global database name, which includes the database name and the domain name. For example, if your database name is `finance` and its domain is `business.com` the service name is `finance.business.com`.

  - **Alternate Servers**: A list of alternate database servers to enable connection failover for the driver. If the primary database server entered as the SID or service name is unavailable, a connection to the servers in this list is attempted until a connection can be established. You can list the servers in SID or service name format, as shown in the following examples:

Using an SID: (`HostName=`*`DB_server_name`*`: PortNumber=1526:SID=ORCL`)

Using a Service Name: (`HostName=`*`DB_server_name`*`: PortNumber=1526:ServiceName=`*`service.name.com`*)

- **TNSNames Connection**: A TNSNames connection uses a `TNSNAMES.ORA` file to retrieve host, port number, and SID information from a server (alias or Oracle net service name) listed in the `TNSNAMES.ORA` file. A TNSNames connection requires the following parameters:
  - **Server Name**: A server name, which is included in a `TNSNAMES.ORA` file included in the TNSNames File field below.
  - **TNSNames File**: The location of your `TNSNAMES.ORA` file. Make sure to enter the entire path to the `TNSNAMES.ORA` file, including the file name itself. You can specify multiple `TNSNAMES.ORA` files.

- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
  - **User Name**: The name of a valid user for the database.

 © 2016, MicroStrategy Inc.

    - **Password**: The password for the user name you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.

- **Notes**: Information to describe this process as part of the workflow.

PostgreSQL

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **PostgreSQL** process to your workflow. The following information is required to create a DSN for PostgreSQL:

- **Data Source Name**: A name to identify the PostgreSQL data source configuration in MicroStrategy. For example, Finance or PostgreSQL-1 can serve to identify the connection.
- **Host Name**: The name or IP address of the machine on which the PostgreSQL database resides. The system administrator or database administrator assigns the host name.
- **Port Number**: The port number for the connection. The default port number for PostgreSQL is usually **5432**. Check with your database administrator for the correct number.
- **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.
- **Default User ID**: The name of a valid user for the PostgreSQL database.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: Tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
    - **Username**: The name of a valid user for the database.
    - **Password**: The password for the default user name that you provided. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Notes**: Information to describe this process as part of the workflow.

Salesforce

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Salesforce** process to your workflow. The following information is required to create a DSN for Salesforce:

- **Data Source Name**: A name to identify the Salesforce data source configuration in MicroStrategy. For example, Finance or Salesforce-1 can serve to identify the connection.
- **Host Name**: The host name to connect to Salesforce.com. You can keep the default value of `login.salesforce.com`.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must supply the following information to test the connection:
  - **Username**: The user name of a user account for Salesforce.com. The user name syntax is *`UserName`*`@salesforce.com`, where *`UserName`* is the specific user account.
  - **Password**: The password for the Salesforce.com user account that was supplied. The password syntax is *`PasswordSecuritytoken`*, where *`Password`* is the password for the user account and *`Securitytoken`* is the additional security token required to access Salesforce.com. Do not use any spaces or other characters to separate the password and security token.

    As part of configuring a connection to your Salesforce.com system, you can include the password and security token as part of the database login, which is a component of a database instance used to access the DSN in MicroStrategy. For steps to create a database login, which you can use to provide the Salesforce.com password and security token, see the *Installation and Configuration Guide*.
- **Notes**: Information to describe this process as part of the workflow.

Sybase ASE

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Sybase ASE** process to your workflow. The following information is required to create a DSN for Sybase ASE:

- **Data Source Name**: A name to identify the Sybase ASE data source configuration in MicroStrategy. For example, Finance or SybaseASE-1 can serve to identify the connection.
- **Network Address**: The network address, in the format *`ServerName_or_IPAddress,PortNumber`*. For example, if your network supports named servers, you can specify an address such as `SybaseASE-1,5000`. You can also specify the IP address such as `123.456.789.98,5000`. Contact your system administrator for the server name or IP address.
- **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.

 © 2016, MicroStrategy Inc.

- **Enable Unicode support (UTF8)**: Select this check box if the database supports UNICODE.
- **Overwrite**: If this check box is selected, the system updates a DSN with the same name with the information provided below. If this check box is cleared and a DSN with the same name exists on the system, no DSN is created and the DSN is not updated.
- **Test Connection**: If this check box is selected, the system tests the DSN information provided to determine if a successful connection can be made. If this check box is cleared, no connection test is performed. If this check box is selected, you must provide the following connection information:
  - **Username**: The name of a valid user for the database.
  - **Password**: The password for the user name that you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Notes**: Information to describe this process as part of the workflow.

Completing a separate System Manager workflow

Rather than include all required processes in a single System Manager workflow, you can group processes into separate workflows. These separate workflows can then be combined in another workflow by including the separate workflows as processes.

By separating tasks into multiple workflows, you can then re-use these workflows as components of other larger workflows. For example, starting Intelligence Server and troubleshooting this service may be required for multiple workflows that you create. You can include the steps to start and troubleshoot Intelligence Server into a separate workflow, and then use this workflow in all the workflows that require these steps.

Once you have created a workflow, you can include it as a configuration in another workflow. In System Manager, from the Connectors and processes pane, add the **Execute System Manager Workflow** process to your workflow. The following information is required:

- **Workflow File**: Click the folder icon to browse to and select a System Manager workflow file. This is the workflow that is included as a process in the current workflow.
- **Starting Process**: Select this check box to specify the first process to attempt for the workflow. Type the name of the process, including the proper case, in the field below. Ensure that the process is enabled as an entry process for the workflow. For steps to enable a process as an entry process, see *Using entry processes to determine the first step in a workflow, page 554*.
- **Use a Parameter File**: Select this check box to specify a parameters file to provide values for the parameters of the workflow. Click the folder icon to browse to and select a parameters file for the workflow. For information on using parameters in a workflow, see *Using parameters for processes, page 638*. You can also specify parameter values using the Use Console Parameters option described below.

© 2016, MicroStrategy Inc.

- **Use a Customized Log File**: Select this check box to specify a log file to save all results of the workflow to. Click the folder icon to browse to and select a log file. This lets you separate the results of each workflow into individual log files. If you clear this check box, the results of the workflow are included in the log file for the main workflow.
- **Use Console Parameters**: Select this check box to manually supply values for parameters of the process. Type the parameters and their values in the field below. If you also use the Use a Parameter File option described above, these values overwrite any values provided in the parameters file. For additional information on how the value of a parameter is determined, see *Using parameters in a workflow, page 641*.
- **Display Output on the Console**: Select this check box to output all results to the System Manager console. If this check box is cleared, the results of any actions taken as part of this System Manager workflow are not displayed on the console and instead only provided in any specified log files.
- Exit code options:
    - **Personalize Success Exit Code(s)**: Select this check box to specify the exit codes that indicate successful execution of the underlying workflow. Type the exit codes in the text box, separating multiple codes with a comma. Valid exit codes must be an integer. The success exit codes you specify here map to a new exit code of 0, which is passed on to the larger workflow to indicate that this workflow executed successfully.
    - **Personalize Failure Exit Code(s)**: Select this check box to specify the exit codes that indicate failed execution of the underlying workflow. Type the exit codes in the text box, separating multiple codes with a comma. Valid exit codes must be an integer. The failure exit codes you specify here map to a new exit code of -1, which is passed on to the larger workflow to indicate that this workflow failed.

  If you do not use the Personalize Exit Code(s) options, or if you configure them incorrectly, one of the following exit codes will be passed on to the larger workflow:
    - **1**: Indicates an undefined execution result, which is treated as a successful execution. This success exit code is passed on to the larger workflow if you do not use the Personalize Exit Code(s) options and the workflow executes, regardless of whether the execution is successful or not.
    - **-2**: Indicates that the input format of the specified exit codes is incorrect, for example, if you use an exit code that is not an integer, or if you separate multiple codes with anything other than a comma.
    - **-3**: Indicates that there is at least one conflict in the personalized exit codes. For example, if you use exit code 4 in both the Success Exit Code(s) list and the Failure Exit Code(s) list.
    - **-5555**: Indicates that the underlying workflow failed to initialize. For example, if the workflow is incomplete, it will not start.
- **Notes**: Information to describe this process as part of the workflow.

 © 2016, MicroStrategy Inc.

Retrieving MicroStrategy information

You can retrieve various information about the MicroStrategy software, which is installed for the machine System Manager is running on, as part of a System Manager workflow. Each MicroStrategy property that you retrieve must be stored in a parameter for the workflow (see *Using parameters for processes, page 638*).

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Retrieve MicroStrategy Properties** process to your workflow. The following information is required to retrieve information on the MicroStrategy installation:

- **MicroStrategy Property**: The information about the system that is retrieved. You can select from the following options:
  - **Home Path**: The path that acts as the home directory for the MicroStrategy installation. This path includes MicroStrategy configuration files that can be modified after a successful installation.
  - **Common Path**: The path that contains important files. The types of files included in this path varies depending on your operating system, but it can include files such as log files, sql files, war files, jar files, libraries, and more.
  - **Build Version**: The build version number of the MicroStrategy software. This version number can be helpful when troubleshooting a MicroStrategy system and when working with MicroStrategy Technical Support.
  - **Release Version**: The major release version of the MicroStrategy software, such as 9.2.1.
- **Parameter**: The System Manager parameter that is used to store the MicroStrategy information that is retrieved.
- **Retrieve this additional property**: Select this check box to retrieve additional information about the MicroStrategy installation. For each of these check boxes that you select, an additional MicroStrategy Property and Parameter pair is made available.
- **Notes**: Information to describe this process as part of the workflow.

Performing system processes

In addition to the various MicroStrategy configurations that can be completed as part of a System Manager workflow, you can also perform various system processes. This lets you perform system processes such as copying, moving, or deleting a file.

You can also execute any process that uses system or third-party tools. This lets you perform custom processes that can be executed from the system's command line.

The system processes that are supported include:

- *Encrypting/Decrypting text or files, page 610*
- *Copying a file or folder, page 611*

© 2016, MicroStrategy Inc.

- *Deleting a file or folder, page 613*
- *Moving a file or folder, page 613*
- *Find and replace information in a file, page 614*
- *Renaming a file or folder, page 616*
- *Unzipping a compressed file, page 616*
- *Compressing files into a zip file, page 616*
- *Downloading files from an FTP or SFTP site , page 617*
- *Uploading files to an FTP or SFTP site , page 618*
- *Executing a SQL statement, page 620*
- *Sending an email, page 621*
- *Delaying a workflow to allow for task completion, page 623*
- *Updating workflow parameters, page 623*
- *Retrieving machine information, page 624*

Encrypting/Decrypting text or files

You can configure a process to encrypt or decrypt specified text or a file.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Cryptographic Service** process to your workflow. The following information is required to perform this process:

- **Action**: Select either Encrypt or Decrypt from the drop-down list. Encrypt algorithmically encodes plain text into a non-readable form. Decrypt deciphers the encrypted text back to its original plain text form.

The Decrypt action only works on text that was encrypted using the Encrypt action. Also, files encoded using the Encrypt action must be decrypted using the Decrypt action. Other encryption/decryption programs will not work.

- **Password**: Select the check box and type the required password if a specific password is required to perform this process. If this option is not selected, it will use the default password specified by System Manager.
- **Text**: Select this option and type the text to be encrypted or decrypted in the text box. This is useful for encrypting or decrypting a small amount of text.
- **File**: Select this option and click the folder icon to select the file to encrypt or decrypt. This option is useful if you have a large amount of text to encrypt or decrypt.
- **Output File**: Click the folder icon to select the file in which to store the encrypted or decrypted results.
- **Overwrite**: Select this check box to overwrite the output file if it already exists.

 © 2016, MicroStrategy Inc.

- **Notes**: Information to describe this process as part of the workflow.

Performing custom processes

You can execute a custom process as part of a System Manager workflow. This can be any process that uses system or third-party tools. However, the process must be executable from the system's command line.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Execute Application** process to your workflow. The following information is required to execute a custom process:

- **Application To Execute**: The command to execute the custom process. This command must meet the syntax requirements of the system it is executed on.
- **Execute In System Shell**: Select this check box to execute the application and any parameters in a Windows command prompt or UNIX shell. If you select this option, the exit code for this process represents the success or failure of creating a new Windows command prompt or UNIX shell. If you clear this option, the exit code for this process represents the success or failure of executing the application, which could fail if an incorrect application name or path is used.
- **Notes**: Information to describe this process as part of the workflow.

Copying a file or folder

You can copy a file or folder as part of a System Manager workflow.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Copy Files** process to your workflow. The following information is required to copy a file or folder:

- **Source File or Directory**: The location of the file or folder to copy. If the path to a file is provided, only that file is copied. If the path to a folder is provided, the folder along with all the files within it are copied. Click the folder icon to browse to and select a file or folder.

  You can also use wildcard characters (* and ?) to select files or folders to copy. For example, you can use the syntax `*.txt` to copy all files with the extension `.txt` in a folder. For additional examples of how you can use these wildcard characters, see *Using wildcard characters in processes, page 643*.
- **Destination File or Directory**: The location of the file or folder to copy the file or folder to.
  - If you are copying a file, you can provide a path to a specific folder location and file name to store the new copy.
  - If you are copying a folder or have used wildcard characters to select multiple files or folders, you can provide a folder location at which to store the files or folders.

© 2016, MicroStrategy Inc.

    - If the location you provide does not exist, a new directory is created with the name of the destination and all source files are copied to the directory. Click the folder icon to browse to and select a file or folder.

- **Overwrite**: If this check box is selected, the system replaces the destination file or folder with the same name as the source file or folder provided. If this check box is cleared and a file or folder with the same name exists on the system, the source file or folder is not copied to the specified location.
- **Notes**: Information to describe this process as part of the workflow.

Creating a file or folder

You can create a file or folder as part of a System Manager workflow.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Create File** process to your workflow. The following information is required to create a file or folder:

- **Select Type**: Determines whether to create a file or folder. Select either **File** or **Directory**.
- **Parent Directory**: The location in which to create the file or folder in. Click the folder icon to browse to and select a folder.
- **File or Directory Name**: The name for the new file or folder:
    - For files, type any file name and extension to create an empty file of that file type. Be aware that this process does not validate whether the file type is valid.
    - For folders, type the folder name. Along with creating a single folder at the parent directory location, you can create a series of subfolders by using backslashes (\). For example, if the parent location is C:\, you can create the following folders:
        - Type `test`. This creates a single folder `C:\test`.
        - Type `test1\test2\test3`. This creates the folder structure `C:\test1\test2\test3`.
- **Notes**: Information to describe this process as part of the workflow.

Determining the number of files in a folder

You can determine the number of files in a folder as part of a System Manager workflow.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Count Files** process to your workflow. The following information is required to determine the number of files in a folder:

- **The Directory**: The location of the top level folder to determine the number of files. Click the folder icon to browse to and select a folder.
- **File Filter**: Select this option to apply a single filter to the files that are to be included in the count of files in a folder. You can then type the filter, including wildcard

 © 2016, MicroStrategy Inc.

characters such as an asterisk (`*`) to represent multiple characters, and a question mark (`?`) to represent a single character. For example, if you type `*.exe`, only files that end with the extension `.exe` are included in the count. If you type `test?.exe`, files such as `test1.exe`, `test2.exe`, `test3.exe`, and `testA.exe` are included in the count.

If you clear this check box, all files in a folder are included in the final count.

- **Among All Files**: Select this option to count files only in the top-level folder.
- **Among All Files and Subfolders Recursively**: Select this option to count files in the top-level folder and all subfolders.
- **Output Parameter**: The number of files in the folder must be stored in a parameter so that it can be passed to another process in the System Manager workflow. Select an output parameter from the drop-down list to store this value.
- **Notes**: Information to describe this process as part of the workflow.

Deleting a file or folder

You can delete a file or folder as part of a System Manager workflow.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Delete Files** process to your workflow. The following information is required to delete a file or folder:

- **File or Directory**: The location of the file or folder to delete. If the path to a file is provided, only that file is deleted. If the path to a folder is provided, the folder and all the files in it are deleted. Click the folder icon to browse to and select a file or folder.

  You can also use wildcard characters (* and ?) to select files or folders for deletion. For example, you can use the syntax `*.txt` to delete all files with the extension `.txt` in a folder. For additional examples of how you can use these wildcard characters, see *Using wildcard characters in processes, page 643*.
- **Notes**: Information to describe this process as part of the workflow.

Moving a file or folder

You can move a file or folder to a new location as part of a System Manager workflow. When a file or folder is moved, the file or folder only exists in the new location provided. This means the file or folder is no longer available in the original location it was moved from.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Move Files** process to your workflow. The following information is required to move a file or folder:

- **Source File or Directory**: The location of the file or folder to move. If the path to a file is provided, only that file is moved. If the path to a folder is provided, the folder along with all the files and folders within it are moved. Click the folder icon to browse to and select a file or folder.

© 2016, MicroStrategy Inc.

You can also use wildcard characters (* and ?) to select files or folders to move. For example, you can use the syntax `*.txt` to move all files with the extension `.txt` in a folder. For additional examples of how you can use these wildcard characters, see *Using wildcard characters in processes, page 643*.

- **Destination File or Directory**: The location of the file or folder to move the file or folder to.
  - If you are moving a file, you can provide a path to a specific folder location and file name to store the file.
  - If you are moving a folder or have used wildcard characters to select multiple files or folders, you can provide a folder location at which to store the files or folders.
  - If the location you provide does not exist, a new directory is created with the name of the destination and all source files will be copied to this directory. Click the folder icon to browse to and select a file or folder.
- **Overwrite**: If this check box is selected, the system replaces the destination file or folder with the same name as the source file or folder provided. If this check box is cleared and a file or folder with the same name exists on the system, the file or folder is not moved to the specified location.
- **Notes**: Information to describe this process as part of the workflow.

Find and replace information in a file

You can search a file for various keywords and phrases, and then replace this information with new content, as part of a System Manager workflow. These changes can be applied by overwriting the file or by creating a new file with all the applicable changes.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Find and Replace File Content** process to your workflow. The following information is required to find and replace content in a file:

- **Source File**: The location of the file to search for content to replace. Click the folder icon to browse to and select a file.
- **Destination File**: The location and name of the file that is created with all content replacements. You can create a new file to retain a copy of the original file, or select the same file as the source file to overwrite the existing file. To overwrite the existing file, you must also select the option Overwrite Destination File If It Already Exists described below. Click the folder icon to browse to and select a file.
- **Overwrite Destination File If It Already Exists**: If this check box is selected, the system replaces the original file with an updated version of the file that has all relevant content updates applied. If this check box is cleared and a file with the same name exists on the system, the file is not updated.
- **Match Case**: If this check box is selected, the system replaces keywords and phrases if the content and the case of the content matches. If this check box is cleared, keywords and phrases are replaced if the content matches, regardless of the case.

 © 2016, MicroStrategy Inc.

- **Keyword**: The keyword or phrase to search for in the file. The search finds and replaces all instances of the keyword or phrase in the file. You must type the keyword or phrase exactly; wildcard characters cannot be used. To replace multiple lines in the file, use `$\n$` to indicate a line break.

  Value: The content used to replace the keyword or phrase. To replace a keyword with multiple lines, use `$\n$` to indicate a line break.

  For example, if you have an XML file that includes multiple instances of the same address, and the person or company with that address has recently moved to another city, you can find and replace all instances of the customer address. If the XML for the address is:

  ```
  <address1>123 Main Street</address1>
  <city>Vienna</city>
  <state>Virginia</state>
  <zip>22180</zip>
  ```

  In the Keyword text box type the following:

  ```
  <address1>123 Main
  Street</address1>$\n$<city>Vienna</city>$\n$<state>Virginia</
  state>$\n$<zip>22180</zip>
  ```

  If the new address should read as follows in the XML

  ```
  <address1>4000 Connecticut Ave NW</address1>
  <address2>Suite 600</address2>
  <city>Washington</city>
  <state>District of Columbia</state>
  <zip>20008</zip>
  ```

  In the Value text box, type the following

  ```
  <address1>4000 Connecticut Ave
  NW</address1>$\n$<address2>Suite
  600</address2>$\n$<city>Washington</city>$\n$<state>Di
  strict of Columbia</state>$\n$<zip>20008</zip>
  ```

- **Use This Additional Keyword / Value Pair**: If this check box is selected, the system includes a find and replace action to search for and replace a given keyword or phrase. Each of these check boxes includes a single, additional find and replace action. For each find and replace action that you include, you must provide the following information:

  - **Keyword**: The keyword or phrase to search for in the file. The search finds and replaces all instances of the keyword or phrase within the file. You must type the keyword or phrase exactly; wildcard characters cannot be used. If you want to replace multiple lines within the file, you can use `$\n$` to indicate a line break.

  - **Value**: The content used to replace the keyword or phrase. If you want to replace a keyword with multiple lines, you can use `$\n$` to indicate a line break.

- **Notes**: Information to describe this process as part of the workflow.

Renaming a file or folder

You can rename a file or folder as part of a System Manager workflow.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Rename Files** process to your workflow. The following information is required to rename a file or folder:

- **Source File or Directory**: The location of the file or folder to rename. Click the folder icon to browse to and select a file or folder.
- **New Name of File or Directory**: The new name for the file or folder.
- **Append Current Date**: Determines whether the current date is automatically added to the end of the new name. The date is added in a YYYY-MM-DD format, such as `NewName-2015-12-21.txt`.
- **Notes**: Information to describe this process as part of the workflow.

Unzipping a compressed file

You can extract the contents of a compressed file as part of a System Manager workflow. The files are extracted to the location that you specify.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Unzip Files** process to your workflow. The following information is required to extract the contents of a compressed file:

- **Zip File**: The location of the compressed file to extract, which can use either zip or gzip format. Click the folder icon to browse to and select a file.
- **Output Directory**: The location of where the files in the compressed file are to be extracted to. Click the folder icon to browse to and select a folder.
- **Overwrite**: Replaces any existing files in the output directory with the files that are being extracted. If this check box is cleared and a file with the same name exists in the output directory, the file is not updated.
- **Notes**: Information to describe this process as part of the workflow.

Compressing files into a zip file

You can compress files and the contents of folders into a zip file as part of a System Manager workflow. The files are extracted to the location that you specify.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Zip Files** process to your workflow. The following information is required to compress files and folders into a zip file:

- **Source File or Directory: The location of the file or folders to include in the zip file. If you select a folder, all of the contents of the folder are included in the zip file, which includes the subfolders and their content. Click the folder icon to browse to and select files and folders.**

 © 2016, MicroStrategy Inc.

You can also use wildcard characters (* and ?) to select files or folders to compress into a zip file. For example, you can use the syntax `*.txt` to select all files with the extension `.txt` in a folder for compression into a zip file. For additional examples of how you can use these wildcard characters, see *Using wildcard characters in processes, page 643*.

- **Output File**: The location and name of the final compressed zip file. Click the folder icon to browse to and select an existing zip file.
- **Operations for Output File**: Determines how an existing zip file is updated. If this check box is cleared and an existing zip file is found, the zip file is not updated and the files are not compressed into a zip file. If you select this check box, you have the following options:
  - **Overwrite**: If an existing zip file is found, the old version is completely replaced by a new zip file.
  - **Append**: If an existing zip file is found, the new files and folders are added to the existing zip file.

    However, if a folder already exists in the same location in the zip file, it is ignored along with any contents of the folder. This means that if a folder has new files, they are not included as part of appending files to the existing zip file.
- **Notes**: Information to describe this process as part of the workflow.

Downloading files from an FTP or SFTP site

You can download files from an FTP or SFTP site as part of a System Manager workflow. These files are downloaded and saved to a folder that you select.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Download using FTP** process to your workflow. The following information is required to download files from an FTP or SFTP site:

- **FTP Server**: The URL for the FTP or SFTP site. You must also define whether the site allows anonymous access or requires a user name and password:
  - **Port Number**: The port number to access the FTP or SFTP site. By default a value of 22 is expected. Select this check box and type the port number for your FTP or SFTP site.
  - **Anonymous**: Defines the connection to the FTP site as anonymous. You cannot use this option if you are connecting to an SFTP site. Type an account for the anonymous connection, such as an email address.
  - **Login**: Defines the connection to the FTP or SFTP site as one that requires a user name and password to log into the FTP or SFTP site. You must provide the following information:
    - **User Name**: The name of a valid user for the FTP or SFTP site.
    - **Password**: The password for the user name that you provided to connect to the FTP or SFTP site. You can use the button to the right of the Password

© 2016, MicroStrategy Inc.

field to determine whether the password characters are shown or asterisks are displayed instead.

  - **Use SFTP**: Encrypts the entire download communication. You must have a secure FTP site for this encryption to work successfully. If you clear this check box, the communication is not encrypted.

If you have both an FTP and an SFTP site, you can choose to clear this check box to use the FTP site, or select this check box to encrypt the communication and use the SFTP site. However, if you only have an FTP site or an SFTP site, your use of this option must reflect the type of site you are using.

- **Download Options**: Determines whether to download a single file or multiple files:
  - **Single File**: Downloads a single file from the FTP or SFTP site. Type the location of the file on the FTP or SFTP site to download.
  - **Multiple Files**: Downloads multiple files from a directory on the FTP or SFTP site. You must provide the following information:
    - **Remote Directory**: The folder within the FTP or SFTP site to download files from.
      - **All Files**: Downloads all the files directly within the folder selected. Subfolders are not downloaded recursively if you select this option.
      - **All Files And Subfolders Recursively**: Downloads all the files and subfolders recursively, within the folder selected.
- **Download To Directory**: The location of the folder to download the files from the FTP site to. Click the folder icon to browse to and select a folder.
- **Overwrite**: If this check box is selected, the system replaces files with the same name as the files or folders downloaded from the FTP or SFTP site. If this check box is cleared and a file or folder with the same name exists on the system, the file or folder is not downloaded from the FTP or SFTP site.
- **Notes**: Information to describe this process as part of the workflow.

Uploading files to an FTP or SFTP site

You can upload files to an FTP or SFTP site as part of a System Manager workflow. These files are uploaded to the FTP or SFTP site that you select.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Upload using FTP** process to your workflow. The following information is required to upload files to an FTP or SFTP site:

- **FTP Server**: The URL for the FTP or SFTP site. You must also define whether the site allows anonymous access or requires a user name and password:
  - **Port Number**: The port number to access the FTP or SFTP site. By default a value of 22 is expected. Select this check box and type the port number for your FTP or SFTP site.

 © 2016, MicroStrategy Inc.

- **Anonymous**: Defines the connection to the FTP site as anonymous. You cannot use this option if you are connecting to an SFTP site. Type an account for the anonymous connection, such as an email address.
- **Login**: Defines the connection to the FTP or SFTP site as one that requires a user name and password to log into the FTP or SFTP site. You must provide the following information:
  - **User Name**: The name of a valid user for the FTP or SFTP site.
  - **Password**: The password for the user name that you provided to connect to the FTP or SFTP site. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
  - **Use SFTP**: Encrypts the entire upload communication. You must have a secure FTP site for this encryption to work successfully. If you clear this check box, the communication is not encrypted.

If you have both an FTP and an SFTP site, you can choose to clear this check box to use the FTP site, or select this check box to encrypt the communication and use the SFTP site. However, if you only have an FTP site or an SFTP site, your use of this option must reflect the type of site you are using.

- **Upload Options**: Determines whether to upload a single file or multiple files:
  - **Single File**: Uploads a single file to the FTP or SFTP site. Click the folder icon to browse to and select a file.
  - **Multiple Files**: Uploads multiple files from a directory to the FTP or SFTP site. You must provide the following information:
    - **Local Directory**: The local folder to upload the files from. Click the folder icon to browse to and select a folder.
      - **All Files**: Uploads all the files directly within the folder selected. Subfolders are not uploaded recursively if you select this option.
      - **All Files And Subfolders Recursively**: Uploads all the files and subfolders recursively, within the folder selected.
- **Upload To Remote Directory**: The location of the folder to upload the files to in the FTP or SFTP site. Type the FTP or SFTP site location.
- **Overwrite**: If this check box is selected, the system replaces files with the same name as the files or folders uploaded to the FTP or SFTP site. If this check box is cleared and a file or folder with the same name exists on the FTP or SFTP site, the file or folder is not uploaded.
- **Notes**: Information to describe this process as part of the workflow.

Executing a SQL statement

You can execute a SQL statement against a database as part of a System Manager workflow. This lets you perform tasks such as updating tables in a database.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Execute SQL** process to your workflow. The following information is required to execute a SQL statement against a database:

- **Connection Information**: Determines whether to connect using a data source name (DSN) or a connection string:
  - **Specify a DSN**: Defines the connection to the database through the use of a DSN. You must provide the following information:
    - **Data Source Name**: The DSN used to access the database.
    - **Authentication for DSN**: Determines if authentication is included as part of the SQL statement. Be aware that some SQL statements can require specific permissions, which means that authentication would be required. Select this check box to authenticate the connection, and supply the following information:
      - **Login**: The name of a valid user for the database.
      - **Password**: The password for the user name that you provided to connect to the database. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
  - **Specify a JDBC Connection String**: Defines the connection to the database through the use of a JDBC connection string. Type a valid connection string in the field provided.
  - **Encoding**: From this drop-down list, select the character encoding for the data source you are connecting to:
    - **Non UTF-8**: Select this option if the data source uses a character encoding other than UTF-8. This can support character encodings such as UTF-16 and USC-2. This encoding option is selected by default.
    - **UTF-8**: Select this option if the data source uses UTF-8 character encoding. For example, Teradata databases may require UTF-8 encoding.
- **Execution**: Determines whether to use a SQL script to supply the SQL statements, or provide a single SQL statement directly in the workflow:
  - **Execute the Contents of an Input File**: Uses a SQL script file to provide the SQL statements. The SQL script file can contain multiple SQL statements to be executed. The syntax of the SQL must be valid for the database it is executed against. Click the folder icon to browse to and select a SQL script file.
  - **Execute a Single SQL Statement**: Lets you type a single SQL statement for execution. The syntax of the SQL must be valid for the database it is executed against, and the statement must end with a semicolon.

 © 2016, MicroStrategy Inc.

- **Save Execution Output Into a File**: If this check box is selected, the system saves all resulting output of executing the SQL statements to the selected file. No output or data is included in the file for SQL statements that do not return any output, such as create table or update table statements. Click the folder icon to browse to and select a file, which can either be a `.txt` or `.csv` file.

  If this check box is cleared, the output of executing the SQL statements is not saved to a file.

- **Include column headers in the output**: Determines whether the column headers are included as part of the SQL statement output. By default, this check box is cleared and the column header information is not included in any output that is saved for the SQL statement. This can be helpful if you plan to use the output of a SQL statement to update the value of a parameter in your System Manager workflow.

  If you select this check box, the column header information is provided in the SQL output along with the associated values. This can provide additional context to the values.

- **Output Parameters**: As part of executing SQL, you can store any results in parameters:

  - **SQL Execution Result**: The resulting output of executing the SQL statements. Select a parameter from the drop-down list to store the SQL result.

- **Notes**: Information to describe this process as part of the workflow.

Sending an email

You can send an email as part of a System Manager workflow. The email can include the results of the workflow, which can provide verification of what processes have been successfully completed.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Send Email** process to your workflow. The following information is required to send an email:

- **From**: The email address of the sender. For an email sent from a System Manager workflow, you must type the email address of the person who deploys the workflow.
- **To**: The email addresses for the intended primary recipients of the email. Use a comma to separate each email address.
- **Cc**: The email addresses of the secondary recipients who should receive a copy of the email addressed to the primary recipients. Select the check box to enter the email addresses. Use a comma to separate each email address.
- **Bcc**: The email addresses of the recipients who should receive the email while concealing their email address from the other recipients. Select the check box to enter the email addresses. Use a comma to separate each email address.
- **Message Subject**: The title of the email that is displayed in the subject line. This can be used to give a brief description of the purpose behind deploying the workflow. Select the check box to enter the message subject.

© 2016, MicroStrategy Inc.

- **Message Body**: The main content of the email. This can give additional details on what was completed as part of the workflow and next steps for a user or administrator to take. Select the check box to enter the message content.
    - **HTML**: Defines the body content of the email to be provided in HTML format. If you clear this check box, the content is provided in plain text format.
- **High Importance**: Defines the email as having high importance. If this check box is cleared, the email is sent without any importance defined for the email.
- **Attach System Manager Log**: If this check box is selected, the system includes the System Manager log file as an attachment to the email. This log file includes all the results of the workflow up to the time of the email request. Any processes in the workflow that are completed after the email request are not included in the log file. If this check box is cleared, the log file is not attached to the email.
- **Attach Any Other File**: If this check box is selected, the system includes a file as an attachment to the email. Click the folder icon to browse to and select a file to include as an attachment. You can also use wildcard characters if the folder or file name is not known when creating the workflow (see *Using wildcard characters in processes, page 643*).

  If you need to send multiple files, you can do one of the following:
    - Compress the required files into a single file such as a `.zip` file. You can include compressing files into a single `.zip` file as part of a System Manager workflow, using the process described in *Compressing files into a zip file, page 616*.
    - Use wildcard characters (* and ?) to select multiple files in a folder. For examples of how you can use these wildcard characters, see *Using wildcard characters in processes, page 643*.
- **Outgoing SMTP Server**: If this check box is selected, the system lets you define the outgoing SMTP server to use to send the email. If this check box is cleared, a default SMTP server is used to send the email. If you choose to specify an SMTP server, you must provide the following information:
    - **SMTP Server**: The SMTP server to use to send the email.
    - You must select the type of port used for the SMTP server. Contact your SMTP server administrator to determine the proper port type:
        - **Plain Text**: Defines the connection to the SMTP sever in plain text, without using any security protocol. By default, this option is selected.
        - **TLS Port**: Defines the connection to the SMTP server as using a Transport Layer Security port.
        - **SSL Port**: Defines the connection to the SMTP server as using a Secure Sockets Layer port.
    - **Port Number**: The port number for the SMTP server.
    - **User Name**: The name of a user account that has the necessary rights to send emails using the SMTP server.

 © 2016, MicroStrategy Inc.

    - **User Password**: The password for the user name that you provided to send emails using the SMTP server. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.

- **Notes**: Information to describe this process as part of the workflow.

Delaying a workflow to allow for task completion

While deploying a System Manager workflow, some processes can take a considerable amount of time. In certain scenarios, your workflow may need these processes to be completed before other processes in the workflow can be started. To support this scenario, you can include a process in your workflow to wait for a specific amount of time.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Wait** process to your workflow. The following information is required to delay the workflow:

- **Waiting Time (sec)**: The number of seconds to remain on the current wait process before proceeding to the next process in a workflow. Type a numeric, integer value to represent the number of seconds to wait before proceeding to the next process in a workflow.

  You can add additional time to the waiting process using the following options:

  You must supply a valid numerical value for the seconds of the wait process, regardless of whether you define the minutes and hours for the wait process. You can type a value of zero (`0`) to define the wait process as a length of time in only minutes and hours.

    - **Minutes**: Select this check box to determine the number of minutes to remain on the current wait process before proceeding to the next process in a workflow. Type a numeric, integer value to represent the number of minutes to wait before proceeding to the next process in a workflow. This time is added to any seconds or hours also defined for the wait process.

    - **Hours**: Select this check box to determine the number of hours to remain on the current wait process before proceeding to the next process in a workflow. Type a numeric, integer value to represent the number of hours to wait before proceeding to the next process in a workflow. This time is added to any seconds or minutes also defined for the wait process.

- **Notes**: Information to describe this process as part of the workflow.

Updating workflow parameters

While deploying a System Manager workflow, you can update the values of parameters that are used in the workflow. Updating parameters during workflow deployment can allow you to react to changes made as part of deploying a workflow. This technique can also be used to help exit a loop in a workflow that is used for troubleshooting purposes, such as checking the availability of an active Intelligence Server.

© 2016, MicroStrategy Inc.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Update Parameters** process to your workflow. The following information is required to update parameters for a workflow:

- **Parameter Name**: The name of the workflow parameter to update.
- **Resolve the value from**: Determines if the parameter value is updated using the contents of a file or a registry. If you clear this check box, the constant value or equation you provide in the New Value field is used to update the parameter. If you select this check box, you must choose one of the following:
  - **File**: Updates the parameter value with the entire contents of a file. If you select this option, you must type the full path to the file in the New Value field. You can use `.txt` or `.csv` files to update the value of a parameter.
  - **Registry**: Updates the parameter value with the value of a registry key. If you select this option, you must type the full path to the registry key in the New Value field.
- **New Value**: The new value to assign to the parameter. If you selected the Resolve the value from check box listed above, you must type the full path to the file or registry key.

  If the **Resolve the value from** check box is cleared, in addition to providing constant values such as integers or strings of characters, you can also use equations to update parameter values. To build these equations, you can include the parameter's value by typing `${ParameterName}`, where *`ParameterName`* is the name of the parameter that you are updating. You can then include any of the arithmetic operators `+`, `-`, `/`, and `*` along with other numeric values. For example, you can create a Loop parameter, and update its value with the following new value equation:

  ```
  ${Loop} + 1
  ```

  it increases the value of the Loop parameter by one each time the Update Parameters configuration is processed in the workflow. This type of parameter value update supports exiting loops in a workflow after a certain number of attempts. For best practices on using the Update Parameters process to support loops in workflows, see *Supporting loops in a workflow to attempt configurations multiple times, page 570*.
- **Update this additional parameter**: Determines if an additional parameter is updated as part of the parameter update process. For each Update this additional parameter check box you select, you must type a Parameter Name and New Value in the respective fields.
- **Notes**: Information to describe this process as part of the workflow.

Retrieving machine information

You can retrieve information about the machine that System Manager is running on as part of a System Manager workflow. Each system property that you retrieve must be stored in a parameter for the workflow (see *Using parameters for processes, page 638*).

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Retrieve System Properties** process to your workflow. The following information is required to retrieve information on the machine:

 © 2016, MicroStrategy Inc.

- **System property**: The information about the system that is retrieved. You can select from the following options:
  - **Operating System Name**: The descriptive name of the operating system, such as Red Hat Enterprise Linux.
  - **Operating System Version**: The version number of the operating system. The version numbering of operating systems varies greatly, so it is important to also know the operating system name along with the operating system version.
  - **User Home Directory**: The path that acts as the current user's home directory, which can be used to store files if other paths are restricted for security reasons.
  - **IP Address**: The IP address of the system, which can be used to connect to the system.
  - **Hostname**: The host name of the system, which can be used to connect to the system.
  - **Java Virtual Machine (JVM) bit-size**: The size allowed for the Java Virtual Machine, which is also often referred to as the heap size. This determines how much memory can be used to perform various Java tasks. You can tune this value to improve the performance of your machine.
  - **Local Machine Date**: The date and time for the system. The time is returned as the time zone for the system. If the time zone for the system is changed, you must restart System Manager to return the new time zone for the machine.
- **Parameter**: The System Manager parameter that is used to store the machine information that is retrieved.
- **Retrieve this additional property**: Select this check box to retrieve additional information about the machine. For each of these check boxes that you select, an additional System property and Parameter pair is made available.
- **Notes**: Information to describe this process as part of the workflow.

Administering cloud-based environments

If your MicroStrategy environment includes cloud-based environments, you can create an Amazon Machine Image (AMI) and get its status. You can also launch, manage, and terminate your cloud-based environments as part of a System Manager workflow.

- System Manager provides support for the following tasks with Amazon cloud-based environments:
  - *Creating an image, page 626*
  - *Getting cloud image status, page 627*
  - *Launching cloud-based environments, page 628*
  - *Managing cloud-based environments, page 629*
  - *Terminating cloud-based environments, page 629*

© 2016, MicroStrategy Inc.

- System Manager provides support for the following tasks with VMware vCloud environments:
  - *Creating a vApp, page 630*
  - *Starting, stopping, and restarting a vApp, page 631*
  - *Starting, stopping, and restarting a virtual machine, page 632*
  - *Duplicating a vApp, page 633*
  - *Deleting a vApp, page 633*
  - *Deleting a virtual machine, page 634*
  - *Creating a virtual machine, page 635*

Creating an image

You can create an Amazon Machine Image (AMI) from an Amazon EBS-backed instance as part of the System Manager workflow. An Amazon Machine Image is a template that contains the software configuration for your server. While creating an image, ensure that the EBS-backed instance is either running or stopped.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Create Image** process to your workflow. The following information is required to create an Amazon Cloud image:

- **Credential Properties File**: The file that includes your secretkey and accesskey for your account. Click the folder icon to browse to and select a credential properties file.
- **Existing Instance ID**: ID of an Amazon EBS-backed instance that is either running or stopped.
- **Name**: Name for the new image.
- **Description**: Description for the new image.
- **Set No Reboot**: Select this check box to prohibit the Amazon EC2 from shutting down the Amazon EBS-backed instance before creating the new image. If you clear this check box, the Amazon EC2 attempts to shut down EBS-backed instance before creating the new image and then restarts the instance.
- **Block Device Mapping**: A block device is a storage device that is physically attached to a computer or accessed remotely as if it were physically attached to the computer. Hard disks, CD-ROM drives, and flash drives are a few examples of block devices.

  A block device mapping defines the block devices to be attached to an AMI. This argument is passed in the form of `devicename=blockdevice`. Where, `devicename` is the name of the device within Amazon EC2 and `blockdevice` can be one of the following:
  - `none`: To omit a mapping of the device from the AMI used to launch the instance, specify `none`. For example: "`/dev/sdc=none`".

 © 2016, MicroStrategy Inc.

- `ephemeralN`: To add an instance store volume to the device, specify `ephemeralN`, where `N` is the volume number. The range of valid volume numbers is `0` to `3`. For example: `"/dev/sdc=ephemeral0"`.
- *`snapshot-id`*`:`*`volume-size`*`:`*`delete-on-termination`*`:`*`volume-type`*`:`*`iops`*

  Note that all of these variables are optional. You can choose to use any or all of them. Refer to your Amazon third-party documentation for additional examples, updates, and information on the block device variables listed below.

  where:

  - *`snapshot-id`* is the ID of the snapshot to use to create the block device. To add an EBS volume (for EBS-backed instance only), specify the snapshot id. For example `"/dev/sdh=snap-7eb96d16"`.
  - *`volume-size`* is the size of the volume in GB. To add an empty EBS volume, omit the snapshot id and specify a volume size. For example `"/dev/sdh=:200"`.
  - *`delete-on-termination`* is to indicate whether the EBS volume should be deleted on termination (`true` or `false`). The default value is `true`. To prevent the volume from being deleted on termination of the instance, specify `false`. For example `"/dev/sdh=snap-7eb96d16::false"`.
  - *`volume-type`*`:`*`iops`* is the volume type (`standard` or `io1`). The default value is `standard`. For example, `"/dev/sdh=:standard"`. To create a provisioned Input/Output Operations Per Second (IOPS) volume, specify `io1` and the number of IOPS that the volume supports. For example `"/dev/sdh= io1:500"`.

- **Output Parameters**: When a cloud-based image is created, various output parameters are provided that include details about the cloud-based environment. It is recommended that you include parameter (see *Using parameters in a workflow, page 641*) for the following output parameter, so that the value can be saved and used for other processes:
  - **New AMI ID**: The newly created image ID for the Amazon Machine Image (AMI).
- **Notes**: Information to describe this process as part of the workflow.

Getting cloud image status

Once your Amazon Cloud image is created, you can determine its state. For example, you can determine if an image is available or has not yet been registered.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Get Image Status** process to your workflow. The following information is required to get the state of your Amazon Cloud image:

- **Credential Properties File**: The file that includes your secretkey and accesskey for your account. Click the folder icon to browse to and select a credential properties file.
- **AMI ID**: The image ID for the Amazon Machine Image (AMI) to use for your cloud-based environment. Type the image ID, which you can retrieve from Amazon's cloud resources.
- **Notes**: Information to describe this process as part of the workflow.

Launching cloud-based environments

You can launch your Amazon cloud-based environments as part of a System Manager workflow.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Launch Instance** process to your workflow. The following information is required to launch a cloud-based environment:

- **Credential Properties File**: The file that includes your secretkey and accesskey for your account. Click the folder icon to browse to and select a credential properties file.
- **AMI ID**: The image ID for the Amazon Machine Image (AMI) to use for your cloud-based environment. Type the image ID, which you can retrieve from Amazon's cloud resources.
- **Instance Type**: The image type for your cloud-based environment, which determines the computing capacity of the cloud-based environment. Select the appropriate instance type from the drop-down list.
- **Zone**: The zone, or network, that the cloud-based environment is launched and deployed to. Type the name for the zone.
- **Key Pair Name**: Select this check box to create the key pair name, which acts as a password to access the cloud-based environment once it is launched. If you clear this check box, this security method is not used with the cloud-based environment.
- **Name Tag**: Select this check box to create a name to distinguish the cloud-based environment. If you clear this check box, no name is provided for the cloud-based environment.
- **Security Group**: Select this check box to create new security groups or use existing security groups. Use a semicolon (;) to separate multiple security groups. If you clear this check box, no security groups are used for the cloud-based environment.
- **Output Parameters**: When a cloud-based environment is launched, various output parameters are provided that include details about the cloud-based environment. It is recommended that you include parameters (see *Using parameters in a workflow, page 641*) for the following output parameters, so that the values can be saved and used for other processes:
  - **Public IP Address**: The public IP address of the cloud-based environment.
  - **Private IP Address**: The private IP address of the cloud-based environment.

 © 2016, MicroStrategy Inc.

    - **Instance ID**: The instance ID of the cloud-based environment. This instance ID is required to terminate a cloud-based environment (see *Terminating cloud-based environments, page 629*).
    - **Public DNS Name**: The public Domain Name System (DNS) name of the cloud-based environment, which is provided upon launching an instance. Using the Amazon EC2 console, you can view the public DNS name for a running instance.
    - **Private DNS Name**: The private Domain Name System (DNS) name of the cloud-based environment, which is provided upon launching an instance. Using the Amazon EC2 console, you can view the private DNS name for a running instance.
- **Notes**: Information to describe this process as part of the workflow.

Managing cloud-based environments

Once your Amazon cloud-based environment is launched, you can start, stop, and force stop the cloud-based environment as part of a System Manager workflow.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Manage Instances** process to your workflow. The following information is required to manage a cloud-based environment:

- **Credential Properties File**: The file that includes your secretkey and accesskey for your account. Click the folder icon to browse to and select a credential properties file.
- **Instance ID**: The instance ID of the cloud-based environment.
- **Action**: The list of actions—that is, start, stop, or force stop—that can be performed on your cloud-based environment. Select the appropriate action from the drop-down list.
- **Output Parameters**: When a cloud-based environment is launched, various output parameters are provided that include details about the cloud-based environment. It is recommended that you include parameters (see *Using parameters in a workflow, page 641*) for the following output parameters, so that the values can be saved and used for other processes:
    - **Public IP Address(es)**: The public IP address of the cloud-based environment.
    - **Private IP Address(es)**: The private IP address of the cloud-based environment.
- **Notes**: Information to describe this process as part of the workflow.

Terminating cloud-based environments

You can terminate your Amazon cloud-based environments as part of a System Manager workflow.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Terminate Instance** process to your workflow. The following information is required to terminate a cloud-based environment:

© 2016, MicroStrategy Inc.

- **Credential Properties File**: The file that includes your secretkey and accesskey for your account. Click the folder icon to browse to and select a credential properties file.
- **Instance ID**: The instance ID of the cloud-based environment.
- **Notes**: Information to describe this process as part of the workflow.

Creating a vApp

You can create a new vApp as part of a System Manager workflow. A vApp is a collection of one or more virtual machines that can be deployed as a single, cloud-based environment.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Create vApp** process to your workflow. The following information is required to create a vApp:

If you are unsure of any of the option values required to create a vApp, contact the vCloud administrator for the necessary information.

- **vCloud Server Name**: The machine name or IP address of a vCloud director server. The syntax for providing a vCloud host name is *`HostName`*`:`*`PortNumber`*, where *`HostName`* is the machine name or IP address, and *`PortNumber`* is the port number for the host.
- **User Name**: The name of a user account that has the necessary rights to work with and create vApps.
- **Login as Administrator**: Select this check box to log in to vCloud as an administrator.
- **Password**: The password for the user name that you provided to create the vApp. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Organization Name**: The organization that authenticates the user.
- **Virtual Datacenter**: The name of the virtual datacenter that allocates the system resources for a vCloud environment.
- **New vApp Name**: The name that is used to identify the vApp.
- **Add VM**: Select this check box to also create a virtual machine for the vApp. If you select this check box, you must provide the following information to create a virtual machine:
    - **Catalog Name**: The name of the catalog that stores the template that you use to create the virtual machine.
    - **Template Name**: The name of the template required to create the virtual machine. A template defines the initial setup and configuration of a virtual machine.
    - **Start the vApp**: Determines if the virtual machine and its associated vApp are powered on so that it can be used after the creation process is completed. Select

 © 2016, MicroStrategy Inc.

this check box to power on the virtual machine and its associated vApp. If you do not select this option, you can use the Manage VM process to power on the virtual machine at a later time (see *Starting, stopping, and restarting a virtual machine, page 632*).

- **Notes**: Information to describe this process as part of the workflow.

Starting, stopping, and restarting a vApp

Once a vApp is created, you can start, stop, and restart the vApp as part of a System Manager workflow. A vApp must be powered on for users to access and work with a vApp. You may need to power off or shut down a vApp to perform various administrative maintenance on the vApp.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Manage vApp** process to your workflow. The following information is required to manage a vApp:

If you are unsure about any of the option values required to manage a vApp, contact the vCloud administrator for the necessary information.

- **vCloud Server Name**: The machine name or IP address of a vCloud director server. The syntax for providing a vCloud host name is `HostName:PortNumber`, where `HostName` is the machine name or IP address, and `PortNumber` is the port number for the host.
- **User Name**: The name of a user account that has the necessary rights to work with vApps.
- **Login as Administrator**: Select this check box to log in to vCloud as an administrator.
- **Password**: The password for the user name that you provided to manage the vApp. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Organization Name**: The organization that authenticates the user.
- **Action**: The type of action to perform on the vApp. Actions performed on a vApp affect the availability of all virtual machines included in the vApp. You can select one of the following actions:
  - **Start**: Starts a vApp so that users can access and work with a vApp.
  - **Stop**: Stops a vApp through a vCloud request, which makes the vApp unavailable to users. This type of vCloud power off request can be monitored by the vCloud system to determine the success or failure of the action.
- **Virtual Datacenter**: The name of the virtual datacenter that allocates the system resources for a vCloud environment.
- **vApp Name**: The name of the vApp to start, stop, or restart.
- **Notes**: Information to describe this process as part of the workflow.

© 2016, MicroStrategy Inc.

Starting, stopping, and restarting a virtual machine

Once a vApp is created, you can start, stop, and restart a virtual machine that is included in a vApp as part of a System Manager workflow. A virtual machine must be powered on for users to access and work with a virtual machine. You may need to power off or shut down a virtual machine to perform various administrative maintenance tasks on the virtual machine.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Manage VM** process to your workflow. The following information is required to manage a virtual machine:

If you are unsure about any of the option values required to manage a virtual machine, contact the vCloud administrator for the necessary information.

- **vCloud Server Name**: The machine name or IP address of a vCloud director server. The syntax for providing a vCloud host name is *`HostName`*`:`*`PortNumber`*, where *`HostName`* is the machine name or IP address, and *`PortNumber`* is the port number for the host.
- **User Name**: The name of a user account that has the necessary rights to work with vApps and virtual machines.
- **Login as Administrator**: Select this check box to log in to vCloud as an administrator.
- **Password**: The password for the user name that you provided to manage the virtual machine. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Organization Name**: The organization that authenticates the user.
- **Action**: The type of action to perform on the virtual machine. You can select one of the following actions:
  - **Power on**: Starts a virtual machine so that users can access and work with the virtual machine.
  - **Power off**: Stops a virtual machine through a vCloud request, which makes the virtual machine unavailable to users. This type of vCloud power off request can be monitored by the vCloud system to determine the success or failure of the action.
- **Virtual Datacenter**: The name of the virtual datacenter that allocates the system resources for a vCloud environment.
- **vApp Name**: The name of the vApp that contains the virtual machine to start, stop, or restart.
- **VM Name**: The name of the virtual machine within the vApp to start, stop, or restart.
- **Notes**: Information to describe this process as part of the workflow.

 © 2016, MicroStrategy Inc.

Duplicating a vApp

You can duplicate a vApp as part of a System Manager workflow. A vApp is a collection of one or more virtual machines, which can be deployed as a single cloud-based environment.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Copy vApp** process to your workflow. The following information is required to duplicate a vApp:

If you are unsure about any of the option values required to duplicate a vApp, contact the vCloud administrator for the necessary information.

- **vCloud Server Name**: The machine name or IP address of a vCloud director server. The syntax for providing a vCloud host name is `HostName:PortNumber`, where `HostName` is the machine name or IP address, and `PortNumber` is the port number for the host.
- **User Name**: The name of a user account that has the necessary rights to work with and create vApps.
- **Login as Administrator**: Select this check box to log in to vCloud as an administrator.
- **Password**: The password for the user name that you provided to duplicate the vApp. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Organization Name**: The organization that authenticates the user.
- **Virtual Datacenter**: The name of the virtual datacenter that allocates the system resources for a vCloud environment.
- **Source vApp Name**: The name of the vApp to duplicate.
- **Destination vApp Name**: The name for the duplicate copy of the vApp.
- **Start the vApp**: Determines if the duplicate copy of the vApp is powered on so that it can be used after the duplication process is completed. Select this check box to power on the vApp. If you do not select this option, you can use the Manage vApp process to power on the vApp at a later time (see *Starting, stopping, and restarting a vApp, page 631*).
- **Notes**: Information to describe this process as part of the workflow.

Deleting a vApp

You can delete a vApp as part of a System Manager workflow. A vApp is a collection of one or more virtual machines, which can be deployed as a single cloud-based environment.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Delete vApp** process to your workflow. The following information is required to delete a vApp:

If you are unsure about any of the option values required to delete a vApp, contact the vCloud administrator for the necessary information.

- **vCloud Server Name**: The machine name or IP address of a vCloud director server. The syntax for providing a vCloud host name is *HostName*:*PortNumber*, where *HostName* is the machine name or IP address, and *PortNumber* is the port number for the host.
- **User Name**: The name of a user account that has the necessary rights to work with and delete vApps.
- **Login as Administrator**: Select this check box to log in to vCloud as an administrator.
- **Password**: The password for the user name that you provided to delete the vApp. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Organization Name**: The organization that authenticates the user.
- **Virtual Datacenter**: The name of the virtual datacenter that allocates the system resources for a vCloud environment.
- **vApp Name**: The name of the vApp to delete.
- **Notes**: Information to describe this process as part of the workflow.

Deleting a virtual machine

You can delete a virtual machine that is part of a vApp as part of a System Manager workflow.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Delete VM** process to your workflow. The following information is required to delete a virtual machine:

If you are unsure of any of the option values required to delete a virtual machine, contact the vCloud administrator for the necessary information.

- **vCloud Server Name**: The machine name or IP address of a vCloud director server. The syntax for providing a vCloud host name is *HostName*:*PortNumber*, where *HostName* is the machine name or IP address, and *PortNumber* is the port number for the host.
- **User Name**: The name of a user account that has the necessary rights to work with and delete virtual machines within vApps.
- **Login as Administrator**: Select this check box to log in to vCloud as an administrator.
- **Password**: The password for the user name that you provided to delete the virtual machine. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Organization Name**: The organization that authenticates the user.

 © 2016, MicroStrategy Inc.

- **Virtual Datacenter**: The name of the virtual datacenter that allocates the system resources for a vCloud environment and includes the vApp that hosts the virtual machine to be deleted.
- **vApp Name**: The name of the vApp that hosts the virtual machine that is to be deleted.
- **VM Name**: The name of the virtual machine to delete.
- **Notes**: Information to describe this process as part of the workflow.

Creating a virtual machine

You can create a new virtual machine and include it in a vApp as part of a System Manager workflow. A vApp is a collection of one or more virtual machines that can be deployed as a single, cloud-based environment.

To perform this configuration, in System Manager, from the Connectors and processes pane, add the **Add VM** process to your workflow. The following information is required to create a virtual machine:

If you are unsure of any of the option values required to create a virtual machine within a vApp, contact the vCloud administrator for the necessary information.

- **vCloud Server Name**: The machine name or IP address of a vCloud director server. The syntax for providing a vCloud host name is *HostName*:*PortNumber*, where *HostName* is the machine name or IP address, and *PortNumber* is the port number for the host.
- **User Name**: The name of a user account that has the necessary rights to work with and create vApps.
- **Login as Administrator**: Select this check box to log in to vCloud as an administrator.
- **Password**: The password for the user name that you provided to create the vApp. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
- **Organization Name**: The organization that authenticates the user.
- **Source**: These options determine if the new virtual machine is created as a duplicate of an existing virtual machine or a new virtual machine is created using a template:
    - **From vApp**: This option duplicates a virtual machine that already exists in the vApp:
        - **Virtual Datacenter**: The name of the virtual datacenter that allocates the system resources for a vCloud environment, and includes the vApp that hosts the virtual machine to be duplicated.
        - **vApp Name**: The name of the vApp that includes the virtual machine to duplicate.

    - **From template**: This option creates a new virtual machine, using a template definition. A template defines the initial setup and configuration of a virtual machine:
        - **Catalog Name**: The name of the catalog that stores the template that you use to create the virtual machine.
        - **Template Name**: The name of the template required to create the virtual machine.
    - **VM Name**: The name of the virtual machine to duplicate from a vApp or create from a template.
- **Destination**: These options determine where the new virtual machine is created:
    - **Virtual Datacenter**: The name of the virtual datacenter that allocates the system resources for a vCloud environment and includes the vApp that will host the new virtual machine.
    - **vApp Name**: The name of the vApp that will host the new virtual machine.
- **Configure New VM**: These options determine additional details about the new virtual machine:
    - **Full Name**: The name for the virtual machine that is created.
    - **Computer Name**: Select this check box to provide the host name of the new virtual machine. If you clear this check box, the name that you specified for Full Name is also used for this host name.
    - **Local Administrator Password**: Select this check box to provide an administrator password for the virtual machine. If you clear this check box, a password is generated or the password in the template used to create the virtual machine is used.
        - **Administrator Password**: The password for the administrator. You can use the button to the right of the Password field to determine whether the password characters are shown or asterisks are displayed instead.
        - **Number of Times to Auto Logon**: The number of times the administrator can start the VM without reentering the login information.
        - **Require Administrator to Change Password on First Login**: Select this check box to require that the Administrator changes the password upon the first login.
    - **Network and IP Assignment**: Select this check box to provide a network name and determine how IP addresses are assigned. This helps to ensure that multiple virtual machines do not use the same IP address, which can cause IP conflict issues in your vCloud system. If you clear this check box, the network and IP assignment configuration is determined by the template used to create the virtual machine.

      When selecting this check box, type the name of the network in the **Network Name** field, and select one of the following IP assignment options:

 © 2016, MicroStrategy Inc.

  - **DHCP**: The IP address is assigned dynamically by a DHCP service on the specified network.
  - **Static IP Pool**: A single, static IP address is allocated automatically from a collection of IP addresses for the network.
  - **Static Manual**: A single, static IP address is allocated. You must type the IP address in the text field. Ensure that the IP address is valid for your network.

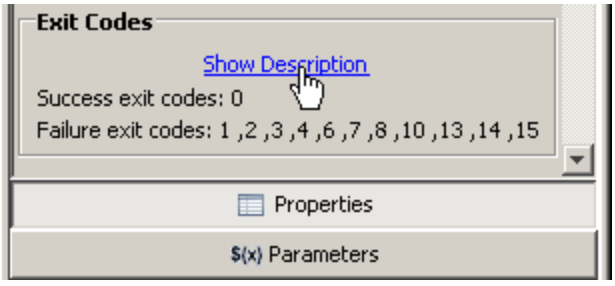
- **Output Parameters**: As part of the virtual machine creation process, you can store important information about the new virtual machine in parameters:
  - **Public IP Address**: The IP address used to access the new virtual machine. Select a parameter from the drop-down list to store the information in that parameter.
  - **Computer Name**: The host name for the new virtual machine. Select a parameter from the drop-down list to store the information in that parameter.
  - **Local Administrator Password**: The administrator password for the virtual machine. Select a parameter from the drop-down list to store the information in that parameter.
- **Notes**: Information to describe this process as part of the workflow.

Determining process resolution using exit codes

System Manager workflows often require information about the resolution of a process to determine the next step to follow in the workflow. An exit code is provided when a process that is part of a System Manager workflow comes to completion. This *exit code* provides information on whether the process was successful.

Along with determining the success or failure of a process, an exit code can also provide additional information on why the process was a success or a failure.

While providing the information for a process, you can review the exit codes for a process. On the Properties pane, scroll down to the bottom and click **Show Description**, as shown in the image below.



Detailed information on each exit code for a process is displayed.

The exit codes for a custom process are dependent on that custom process. Refer to any documentation related to the custom process to determine possible exit codes.

You can use these exit codes to determine the next step to take in a workflow:

- Using the success and failure connectors lets you guide the workflow based on whether the process was completed with a success or failure exit code. For additional information on how connectors determine the logical order of a workflow based on the exit code of the process they are coming from, see *Using connectors to create the logical order of a workflow, page 553*.
- Using a decision process, you can guide the workflow according to error codes rather than just whether the process was considered successful or unsuccessful. This can help to support additional troubleshooting and error checking during a workflow. For examples of how decisions can be used to guide a workflow on more than just the success or failure of a process, see *Using decisions to determine the next step in a workflow, page 555*.

Using parameters for processes

While all the necessary configuration information can be provided for each process, some scenarios require that the details about the process be provided when the workflow is executed. This can be required for the scenarios listed below:

- Storing user credentials within System Manager introduces a security risk.
- Configuration information is not known until or during the actual configuration.

To provide a flexible solution to these types of problems, System Manager lets you define parameters as part of your workflow. These parameters can be used to define configuration information for the processes supported by System Manager. The values of these parameters can be provided as part of the workflow, as part of a parameters file to execute the workflow, and as input from the user performing the workflow from the command line.

Creating parameters for a workflow

A workflow has one set of parameters that is shared for all processes. The parameters that are created for a workflow can be used in any configuration task that can accept parameters as values in a process. Parameters can also be used in decisions in a workflow.

The steps below show you how to create parameters for a workflow.

To create parameters for a workflow

This procedure assumes you are creating new parameters for a workflow. For information on importing parameters for a workflow, see *Importing parameters into a workflow, page 639*.

1. Open System Manager.
   - To open System Manager in a Windows environment:

 © 2016, MicroStrategy Inc.

a From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **System Manager**.

- To open System Manager in a UNIX or Linux environment:

  a From a UNIX/Linux console window, browse to *HOME_PATH*, where *HOME_PATH* is the directory that you specified as the home directory during installation.

  b Browse to the folder `bin`.

  c Type `mstrsysmgrw`, and then press ENTER.

The System Manager home page is displayed.

2 Expand the **Properties and parameters** pane on the right side of System Manager, and click **Parameters** near the bottom.

3 Click **Add new parameter** (displayed as a green plus symbol) to create a new parameter. Name and Value fields are displayed.

4 Type the following information:

- **Name**: The name for the parameter. This is the name that is used to identify the parameter in a process or decision within the workflow.
- **Value**: The value that is used in place of the parameter when the workflow is executed. This works as the default value for the parameter if no value for the parameter is given from the command line when the workflow is executed. For information on the precedence of providing values for parameters, see *Providing parameter values during deployment of a workflow, page 642*.

If the parameter provides sensitive information such as user passwords, you can leave the value blank. However, be aware that these parameters must be provided a value when the workflow is executed.

- **Confidential**: Select the check box to turn off any logging and feedback information for parameter values that are updated by a process in your workflow (defined as an output parameter of a process). For example, if you save the result of a SQL execution to a parameter, this result is hidden from any System Manager logs. If the parameter value for a confidential parameter has to be shown in the feedback console, it is displayed as asterisks instead of the actual value. For information on the feedback console, see *Using System Manager to test and deploy a workflow, page 644*.

5 Once a parameter is created in a workflow, you can use it in a workflow, as described in *Using parameters in a workflow, page 641*. You can also use the Update Parameters process (see *Updating workflow parameters, page 623*) to update the value of a parameter during the deployment of a workflow.

Importing parameters into a workflow

You can import parameters into a workflow that have been saved as a parameters response file. This lets you update the values for your workflow.

© 2016, MicroStrategy Inc.

When parameters are imported into a workflow, any existing parameters are updated with the values included in the parameters file. Parameters can only be updated when importing a parameters file. This means that if a parameter does not already exist in a workflow, it is not created when importing the parameters file.

Additionally, if parameters are in the workflow that are not defined in the parameters file, the value for the parameters is not updated during the import process.

Prerequisites

- The workflow you are importing parameters for already has parameters defined for it. Only these parameters can be updated by importing a parameters file.

To import parameters into a workflow

1. Open System Manager.
   - To open System Manager in a Windows environment:
     a. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **System Manager**.
   - To open System Manager in a UNIX or Linux environment:
     a. From a UNIX/Linux console window, browse to *HOME_PATH*, where *HOME_PATH* is the directory that you specified as the home directory during installation.
     b. Browse to the folder `bin`.
     c. Type `mstrsysmgrw`, and then press ENTER.

   The System Manager home page is displayed.
2. Expand the **Properties and parameters** pane on the right side of System Manager, and click **Parameters** near the bottom.
3. From the **Workflow** menu, select **Import Parameter File**. The Import from dialog box opens.
4. Select the parameters file to import and click **Open**. You are returned to System Manager and the parameters are updated accordingly. If the changes are not what you expected, you can click **Clear** to undo all the parameter updates.

Exporting parameters to a file

You can export the parameters in a workflow to a file. This file can serve various purposes:

- You can import parameters into other workflows.
- You can modify the parameter file and apply updates to the original workflow.

 © 2016, MicroStrategy Inc.

- You can modify the parameter file and include it during execution to make changes just before execution.
- You can modify the parameter file to include comments, which can provide additional information on the parameters and their values. To include a comment in a parameters file you can use the characters `//` or `#` to denote a line in the parameters file as a comment. Any line that begins with either `//` or `#` is ignored when using the parameters file with System Manager.

The steps below show you how to export the parameters of a workflow to a file.

To export parameters of a workflow to a file

1. Open System Manager.
   - To open System Manager in a Windows environment:
     a. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **System Manager**.
   - To open System Manager in a UNIX or Linux environment:
     a. From a UNIX/Linux console window, browse to *HOME_PATH*, where *HOME_PATH* is the directory that you specified as the home directory during installation.
     b. Browse to the folder `bin`.
     c. Type `mstrsysmgrw`, and then press ENTER.

   The System Manager home page is displayed.
2. From the **Workflow** menu, select **Export Parameter File**. The Export to dialog box opens.
3. In the **File name** field, type a name for the parameters file.
4. Select a location to save the file and click **Save**.

Using parameters in a workflow

Parameters can be used in processes or decisions of a workflow to provide flexibility as to when the information is provided.

Parameters can be included in any option that takes some type of text or numeric data as input. For example, a Password field can take a parameter that supplies a password to access the task or system resource for a process. However, check boxes and any other options that do not accept text or numeric data cannot use parameters.

To use a parameter in a process or decision, you must use the following syntax:

```
${ParameterName}
```

In the syntax listed above, `ParameterName` is the name of the parameter. During execution, this is replaced with the value for the parameter.

The values for parameters can be provided in a few different ways. For information on how parameter values can be provided and the precedence of each option, see *Providing parameter values during deployment of a workflow, page 642* below.

Providing parameter values during deployment of a workflow

The value for a parameter can be provided in the following ways:

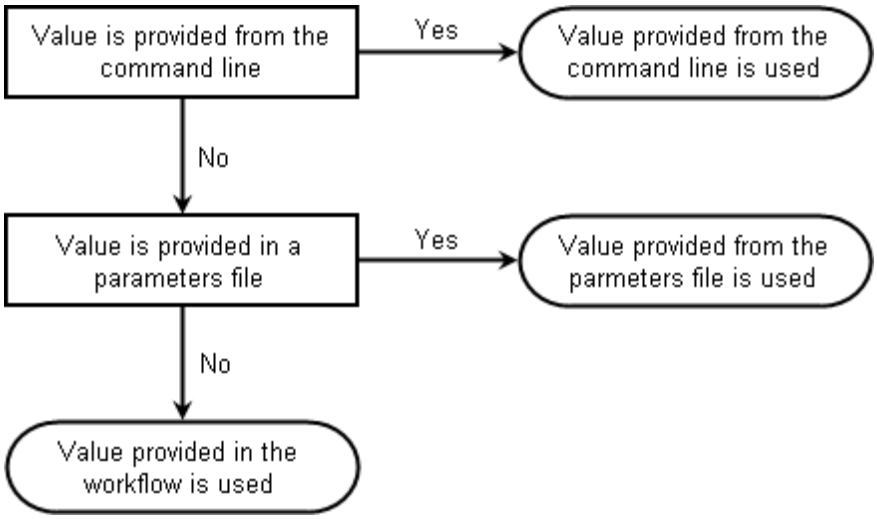
- When defining the parameters for the workflow. These values act as the default value of the parameter.
- In a parameters file. This file can be used during the execution of a workflow to provide updated values for the parameters.
- From the command line during execution of a workflow. This lets the user executing the process provide sensitive information such as user passwords on the command line rather than saving them in a workflow.
- You can also use the Update Parameters process (see *Updating workflow parameters, page 623*) to update the value of a parameter during the deployment of a workflow.

When a workflow is executed, parameters are replaced with their respective values, as described below:

- If the value for a parameter is provided from the command line during execution, this value is used. Any values for the parameter provided in a parameters file or default values provided in the workflow are ignored.
- If the value for a parameter is not provided from the command line during execution, but a value for the parameter is provided in a parameters file, the value from the parameters file is used. The default value provided in the workflow is ignored.
- If the value for a parameter is not provided in a parameters file or from the command line during execution, the default value provided when defining a parameter in a workflow is used.

The image below summarizes how the value of a parameter is determined when executing a workflow:

 © 2016, MicroStrategy Inc.



Using wildcard characters in processes

System Manager allows you to use wildcard characters to provide configuration information for some of the processes possible with a System Manager workflow. Using wildcard characters to provide configuration information for processes in a System Manager workflow can allow you to:

- Refer to folders or files that do not exist yet or do not have known names. For example, a file or folder can be created as part of the same System Manager workflow. If the full name of the file or folder is not known (for example, the file name itself might include creation time information) you can use wildcard characters to refer to the expected file or folder.
- Select multiple files for a single process, such as attaching multiple files to an email. For example, rather than simply listing a single file, you can use wild cards to select all `.txt` files in a folder.

System Manager processes that support wild cards as part of their configuration include:

- Sending an email (see *Sending an email, page 621*)
- Deleting files or folders (see *Deleting a file or folder, page 613*)
- Moving files (see *Moving a file or folder, page 613*)
- Copying files (see *Copying a file or folder, page 611*)
- Compressing files into a zip file (see *Compressing files into a zip file, page 616*)

For the configurations of a System Manager process that can use wildcard characters, the following characters are supported:

- The * (asterisk) character: You can use * to represent one or more characters. Some examples of how you can use this wildcard character include:
  - `*.txt`

    This syntax would search for and select all .txt files in a given folder.

    - *filename*.*

      This syntax would search for and select all files, regardless of file extension, with the name *filename*.

    - *.*

      This syntax would select all files in a given folder.

    - *

      This syntax would search for and select all files and folders in a given folder.

- The ? (question mark) character: You can use ? to represent any single character. Some examples of how you can use this wildcard character include:

    - *filename*?.ini

      This syntax would search for and select all .ini files with the name *filename* and a single character. For example, the syntax config?.ini would select files such as config1.ini, configA.ini, and so on.

    - *filename*.??

      This syntax would search for and select all files with the name *filename* and any two character file extension.

You can also use a combination of both * and ? wildcard characters.

Deploying a workflow

Once you create a workflow, you can deploy the workflow to attempt the processes that are included in the workflow. System Manager provides the following methods for deploying a workflow:

- *Using System Manager to test and deploy a workflow, page 644*: System Manager's interface can be used to test and deploy a workflow.
- *Using the command line to deploy a workflow, page 647*: System Manager's command line version can be used to deploy a workflow without the use of an interface. This can be beneficial for silent configuration routines and OEM deployments.

Using System Manager to test and deploy a workflow

Once you create a workflow using System Manager, you can use the same System Manager interface to test and deploy a workflow.

 © 2016, MicroStrategy Inc.

Be aware that some processes are dependent on the machine that you use to deploy the workflow. For example, if you include processes to create DSNs, the DSNs are created on the machine that you use to deploy the workflow.

The steps below show you how to deploy a workflow from within System Manager.

Prerequisites

- You have created a workflow and saved it in a location that can be accessed from the machine that you are deploying the workflow on. Steps to create a workflow are provided in *Creating a workflow, page 546*.
- System Manager is installed. This tool is installed as part of the general MicroStrategy product suite.
- You have installed any MicroStrategy products and components that are required for the processes of a workflow. For the products and components required for each process, see *Defining processes, page 578*.
- If required, you have created a parameters file to provide values for the parameters of the workflow and saved it in a location that can be accessed from the machine that you are deploying the workflow on.

To deploy a workflow using System Manager

1. Open System Manager.
   - To open System Manager in a Windows environment:
     a. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **System Manager**.
   - To open System Manager in a UNIX or Linux environment:
     a. From a UNIX/Linux console window, browse to *HOME_PATH*, where *HOME_PATH* is the directory that you specified as the home directory during installation.
     b. Browse to the folder `bin`.
     c. Type `mstrsysmgrw`, and then press ENTER.

   The System Manager home page is displayed.
2. From the **File** menu, select **Open Workflow**. The Open workflow dialog box is displayed.
3. Browse to the workflow file, select the file, and then click **Open**. The workflow is displayed within System Manager.
4. If you need to supply values for the parameters in the workflow by importing a parameters file, perform the steps provided in *Importing parameters into a workflow, page 639*.
5. From the **View** menu, select **Options**. The Options dialog box opens.

6 In the **Log file path** field, type the path of a log file or use the folder (browse) icon to browse to a log file. All results of deploying a workflow are saved to the file that you select.

7 In the **Maximum Concurrent Threads** field, type the maximum number of tasks that can processed at the same time. This ensures that even if a workflow requests a certain number of tasks to be processed at the same time, only the specified limit is allowed to run at the same time. The default value for this option is either the number of CPUs for the current system, or 2, whichever value is greater. For information on creating workflows that execute multiple tasks at the same time and how to limit the number of simultaneous tasks, see *Processing multiple tasks simultaneously, page 562* and *Limiting the number of parallel tasks to prevent over consumption of system resources, page 565* respectively.

8 Click **OK** to save your log file selection and close the Options dialog box.

9 From the **Workflow** menu, point to **Execute Workflow**, and then select **Run Configuration**. The Run Configuration dialog box opens.

You can execute a single process in a workflow to test the process, or to perform the process separately. To execute a single process, right-click the process and select **Execute Process**.

10 From the **Starting process** drop-down list, select the process to act as the first process in the workflow. You can only select processes that have been enabled as entry processes for the workflow.

11 In the **Parameters** area, type any parameters required to execute the processes in the workflow, which can include user names, passwords, and other values. To include multiple parameter and value pairs, you must enclose each parameter in double quotes (" ") and separate each parameter and value pair using a space. The following example contains the syntax to provide values for the parameters `UserName` and `Password`:

```
"UserName=User1" "Password=1234"
```

For information on supplying parameters for a workflow, see *Using parameters for processes, page 638*.

12 Click **Run** to begin the workflow. As the workflow is being executed the results of each process are displayed in the Console pane. You can use the Console pane to review additional details on the results of each process and export these details. The results are also saved to the log file that you specified earlier. If you marked any process parameters as Confidential, the parameter value will either not be displayed in the feedback console and logs, or it will be masked and displayed as asterisks instead of the actual value.

If you need to end the workflow prematurely, from the **Workflow** menu, select **Terminate Execution**. A dialog box is displayed asking you to verify your choice to terminate the execution of the workflow. To terminate the execution of the workflow, click **Yes**. If some processes in the workflow have already been completed, those processes are not rolled back.

 © 2016, MicroStrategy Inc.

Using the command line to deploy a workflow

Once you create a workflow using System Manager, you can use a command line version of System Manager interface to deploy a workflow. The command line version lets you deploy a workflow without having to use an interface, which may be useful for silent configuration routines and OEM deployments.

Be aware that some processes are dependent on the machine that you use to deploy the workflow. For example, if you include processes to create DSNs, the DSNs are created on the machine that you use to deploy the workflow.

The command line version of System Manager is a one-line command line tool. This means that the command to begin the deployment is included in a single statement. The syntax of the statement depends on the environment you are deploying the workflow on:

- Windows: `MASysMgr.exe`, followed by the parameters listed below.
- UNIX and Linux: `mstrsysmgr`, followed by the parameters listed below.

Of the parameters listed below, only `-w` to specify a workflow file is required; all other parameters are optional:

- `-w "`*`WorkflowFile`*`"`: This parameter is required to specify the workflow to deploy. *`WorkflowFile`* is the path to the workflow file. For example, `-w "C:\Create DSNs.smw"` is valid syntax to deploy the `Create DSNs.smw` workflow file in a Windows environment.
- `-s "`*`EntryProcess`*`"`: This parameter can be used to specify the first process to attempt for the workflow. Only processes that have been enabled as entry processes (see *Using entry processes to determine the first step in a workflow, page 554*) can be used as the first process in a workflow. *`EntryProcess`* is the name of the process as it is defined in the workflow.
- `-f "`*`ParametersFile`*`"`: This parameter can be used to specify a parameters file, which supplies values for the parameters in the workflow. *`ParametersFile`* is the path to the parameters file. For example, `-f "C:\Parameters.smp"` is valid syntax to use the `Parameters.smp` parameter file in a Windows environment. For information on creating a parameters file, see *Exporting parameters to a file, page 640*.
- `-l "`*`LogFile`*`"`: This parameter can be used to specify a log file. All results of deploying a workflow are saved to the file that you specify. *`LogFile`* is the path to the log file. For example, `-l "C:\Workflow Results.log"` is valid syntax to use the `Workflow Results.log` log file in a Windows environment.
- `-showoutput`: This parameter can be used to display all the results of deploying the workflow to the command line. If you are deploying a workflow as a completely silent process, excluding this option prevents these results from being displayed on the command line. The results can still be retrieved from the log file after deployment is complete.
- `-p "`*`ParameterName1=Value1 ParameterName2=Value2`*`"`: This parameter can be used to specify values for parameters of the workflow. Any parameter values that are provided in this way are used in place of values provided in the workflow itself, as well as provided through a parameters file. Providing parameter values

© 2016, MicroStrategy Inc.

directly during command line execution is often required to supply login and password credentials specific to the machine or user environment for a given deployment.

To include multiple parameter and value pairs, you must enclose each parameter in double quotes (" ") and separate each parameter and value pair using a space. For example, `-p "UserName=User1" "Password=1234"` is valid syntax to provide values for the parameters `UserName` and `Password`.

The steps below show you how to deploy a workflow using the command line version of System Manager.

Prerequisites

- You have created a workflow and saved it in a location that can be accessed from the machine that you are deploying the workflow on. You have created a workflow. Steps to create a workflow are provided in *Creating a workflow, page 546*.
- System Manager is installed. This tool is installed as part of the general MicroStrategy product suite.
- You have installed any MicroStrategy products and components that are required for the processes of the workflow. For the products and components required for each process, see *Defining processes, page 578*.
- If required, you have created a parameters file to provide values for the parameters of the workflow and saved it in a location that can be accessed from the machine that you are deploying the workflow on.

To deploy a workflow using the command line version of System Manager

1. Open a command line.
2. Check to verify that System Manager is installed on the machine:
   - Windows: Type `MASysMgr.exe` and press `ENTER`.
   - UNIX and Linux: Type `mstrsysmgr` and press `ENTER`.

   If help information for using the command line version of System Manager is displayed, this means that System Manager is installed correctly.
3. Type the command to deploy the workflow:
   - Windows: Type `MASysMgr.exe` and include the parameters listed above in *Using the command line to deploy a workflow, page 647* as required. For example, the command below is a valid command to deploy a System Manager workflow on a Windows environment:

     ```
     MASysMgr.exe -w "C:\Create DSNs.smw" -s "Create Oracle
     DSN" -f "C:\Parameters.smp" -l "C:\Workflow Results.log" -
     showoutput -p "UserName=User1" "Password=1234"
     ```

 © 2016, MicroStrategy Inc.

- UNIX and Linux: Type `mstysysmgr` and include the parameters, listed above in *Using the command line to deploy a workflow, page 647*, as required. For example, the command below is a valid command to deploy a System Manager workflow on a UNIX or Linux environment:

```
mstrsysmgr -w "$HOME/Create DSNs.smw" -s "Create Oracle
DSN" -f "$HOME/Parameters.smp" -l "$HOME/Workflow
Results.log" -showoutput -p "UserName=User1"
"Password=1234"
```

4 Once you have typed the full command, press `ENTER`. The workflow is started and results are saved to the log file, as well as displayed on the screen if you included the parameter `-showoutput`.

Supporting a silent deployment with the command line

The command line version of System Manager lets you support silent and OEM deployments of your workflows. You can support silent and OEM deployments of System Manager using the techniques listed below:

- Ensure that the machine that is to be used for the deployment meets all the prerequisites listed in *Using the command line to deploy a workflow, page 647*.
- Determine the syntax to deploy the workflow using the command line version of System Manager. The required and optional parameters are described in *Using the command line to deploy a workflow, page 647*. This syntax can then be used in one of the following ways:
  - Log in to the machine to perform the deployment from, and use the steps provided in *To deploy a workflow using the command line version of System Manager, page 648* to deploy the workflow.
  - Send the required syntax to the user or administrator of the machine to perform the deployment from. Along with the required syntax, provide information on the parameters that the user needs to provide in the command line request. This user can then follow the steps provided in *To deploy a workflow using the command line version of System Manager, page 648* to deploy the workflow.
- Review the results of the deployment using the log file specified to verify that the required processes were completed successfully.

 © 2016, MicroStrategy Inc.

15

AUTOMATING ADMINISTRATIVE TASKS WITH COMMAND MANAGER

Introduction

MicroStrategy Command Manager lets you perform various administrative and application development tasks by using text commands that can be saved as scripts. You can manage configuration settings in the MicroStrategy platform for either project sources or Narrowcast Server metadatas. With Command Manager you can change multiple configuration settings all at once, without using the Developer or Narrowcast Administrator interface. You can also create scripts to be run at times when it would not be convenient for you to make the changes.

The Command Manager script engine uses a unique syntax that is similar to SQL and other such scripting languages. For a complete guide to the commands and statements used in Command Manager, see the Command Manager Help.

Using Command Manager

With Command Manager you can change multiple configuration settings all at once as part of an automated script. For example, you can change the system to allow more low

© 2016, MicroStrategy Inc.

priority jobs to complete at night than during regular hours. To do this, you could create a script to increase the number of low priority database connections and modify several Intelligence Server governor settings. Then, you could schedule the script to run at 8 P.M. You could then create another script that changes the database connections and Intelligence Server settings back for daytime use, and schedule that script to run at 6 A.M.

To schedule a script to run at a certain time, use the Windows AT command with the `cmdmgr` executable. For the syntax for using the executable, see *To invoke Command Manager from another application, page 657*.

Here are more examples of tasks you can perform using Command Manager:

- User management: Add, remove, or modify users or user groups; list user profiles
- Security: Grant or revoke user privileges; create security filters and apply them to users or groups; change security roles and user profiles; assign or revoke ACL permissions; disconnect users or disable their accounts
- Server management: Start, stop, or restart Intelligence Server; configure Intelligence Server settings; cluster Intelligence Server machines; change database connections and logins; manage error codes and customize output data; disconnect active sessions on server or project
- Database management: create, modify, and delete connections, connection mappings, logins, and database instances
- Project management: List or kill jobs; change a project's mode (idle, resume); expire and delete caches; change filter or metric definitions; manage facts and attributes; manage folders; update the project's schema; manage shortcuts; manage hidden properties; create tables and update warehouse catalog tables
- Scheduling: Trigger an event to run scheduled reports
- Narrowcast Server administration: Start and stop a Narrowcast Server; start, stop, and schedule Narrowcast Server services; add, modify, and remove subscription book users; define and remove user authentication

Privileges required for using Command Manager

Any users who want to use Command Manager must have the Use Command Manager privilege. In addition, they must have the usual privileges for any system maintenance tasks they want to perform. For example, to modify the number of low priority database connections, the user must have the Create And Edit Database Instances And Connections privilege.

A common way to delegate administrative tasks that can be performed with Command Manager is to grant a user the Use Command Manager privilege along with one or more security roles. The user can then perform all tasks related to that security role and is prohibited from performing other tasks.

 © 2016, MicroStrategy Inc.

For full access to all Command Manager functionality, a user must have all privileges in the Common, Distribution Services, and Administration groups, except for Bypass All Object Security Access Checks.

Creating and executing scripts

From the Command Manager graphical interface, you can create and execute Command Manager scripts. The script editor has many of the same features as a standard text editor, with copy/paste and one-level undo functionality. Other features of the script editor include a script syntax checker, color-coded script syntax (see *Color-coding the text in a script, page 653*), and sample script outlines (see *Script outlines, page 654*).

Command Manager also includes a command line interface for use in environments that do not support the graphical interface, such as certain UNIX/Linux shell environments, or terminal connections. For instructions on using the Command Manager command line interface, see *Using Command Manager from the command line, page 663*.

To start the Command Manager graphical interface

In Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then choose **Command Manager**.

In UNIX/Linux: Browse to the MicroStrategy Home folder, then to the `/bin` subfolder. Type `mstrcmdmgrw` and press **ENTER**.

For more information about using Command Manager and for script syntax, see the Help. To access the help in the graphical Command Manager interface, click the **Help** menu, then select **Command Manager Help**.

Color-coding the text in a script

The Command Manager script editor can display color-coded text according to its function in the script or procedure.

In a Command Manager script:

- Reserved words display as blue.
- Words or phrases in quotation marks display as gray.
- Numbers display as red. Dates display as red with blue slashes.
- GUIDs display as green.
- All other text appears in black.

In a Command Manager procedure:

© 2016, MicroStrategy Inc.

- Keywords, such as `if` or `boolean`, display as purple, and bold.
- Functions, classes, and methods display as red.
- Command Manager statements display as blue.
- Comments display as green.
- All other text appears in black.

Script outlines

The Command Manager script outlines help you insert script statements with the correct syntax into your scripts. Outlines are preconstructed statements with optional features and user-defined parameters clearly marked.

Outlines are grouped by the type of objects that they affect. The outlines that are available to be inserted depend on whether the active Script window is connected to a project source or a Narrowcast server. Only the outlines that are relevant to the connected metadata source are available.

To insert an outline into a script

1. Start the Command Manager graphical interface.
2. Connect to a metadata source. A new script opens automatically when you connect to a source.
3. From the **Edit** menu, select **Insert Outline**. The Choose Outline dialog box opens.
4. Navigate the Outline tree to locate the outline you want, and select it.
5. Click **Insert** to place the selected outline into the script.
6. Click **Cancel** to close the Choose Outline dialog box.
7. Modify the script as needed.

Procedures in Command Manager

Command Manager procedures are reusable scripts that can be executed from other scripts. You can reuse procedures with different input values, so that the procedure performs the same task in a slightly different way. Procedures can use Command Manager syntax, or they can be written in the Java programming language and incorporate Command Manager statements in Java commands.

For example, you can create a procedure called NewUser that creates a user and adds the user to groups. You can then call this procedure from another Command Manager script, supplying the name of the user and the groups. To use the procedure to create a user named KHuang and add the user to the group Customers, use the following syntax:

```
EXECUTE PROCEDURE "NewUser" ("KHuang", "Customers");
```

 © 2016, MicroStrategy Inc.

where NewUser is the name of the procedure, and KHuang and Customers are the inputs to the procedure.

Procedures are available only for use with project sources. Procedures cannot be used with Narrowcast Server statements.

Command Manager contains many sample procedures that you can view and modify. These are stored in the Command Manager directory, in `\Outlines\Procedure_Outlines\Sample_Procedures\`

For instructions on how to use procedures, see the Command Manager Help. (From within the Command Manager graphical interface, press **F1**.)

Using Java in Command Manager procedures

Java is a simple yet powerful programming language that is widely used in the software industry. Java can be integrated into Command Manager procedures to automate repetitive tasks such as creating multiple users, or recursively listing all the folders in a project. Java is supported in Command Manager out-of-the-box; no additional software must be installed to execute Java commands.

To include Java in a Command Manager script, you write a procedure containing the Java code, and execute the procedure from a Command Manager script. Java cannot be included directly in a Command Manager script. For detailed instructions on using Java in procedures, see the Command Manager Help. (From within the Command Manager graphical interface, press **F1**.)

Java is supported only in procedures, and procedures are supported only with project sources. Java commands cannot be used in scripts to be executed against a Narrowcast Server metadata.

Do not use the `System.exit` command to exit a procedure. This command terminates the entire Command Manager process.

Command Manager provides two special commands that can be used by Java scripts to execute Command Manager commands:

- `execute` runs any Command Manager command, but it does not return the results.
- `executeCapture` runs any Command Manager command and returns the results in a ResultSet object. This object behaves like a standard ResultSet object in Java: you can iterate through the results and retrieve individual items, which can then be used to extract properties of the results. This enables you to use the results elsewhere in the procedure.

  For a detailed list of the ResultSet columns used in each Command Manager LIST statement, see the statement syntax guide for that statement in the Command Manager Help.

© 2016, MicroStrategy Inc.

Executing a Command Manager script

You can execute Command Manager scripts in the following ways:

- From the Command Manager graphical interface (see *Creating and executing scripts, page 653*)
- From the Command Manager command line interface (see *Using Command Manager from the command line, page 663*)
- Invoke the Command Manager executable, including necessary parameters such as the script file to run, from the Windows scheduler, Windows command prompt, or other applications such as system management software.

Command Manager Runtime is a lightweight version of Command Manager for bundling with OEM applications. Command Manager Runtime has fewer execution options and supports fewer statements than Command Manager. For more information about Command Manager Runtime, see *Using Command Manager with OEM software, page 663*.

Command Manager does not automatically lock a project or configuration when it executes statements. To avoid metadata corruption, use the LOCK PROJECT or LOCK CONFIGURATION statements in any Command Manager scripts that make changes to a project or server configuration. For more information about locking and unlocking a project or configuration, see *Project and configuration locking, page 659*.

To execute a script from the Command Manager graphical interface

1. Start the Command Manager graphical interface:
   - In Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then choose **Command Manager**.
   - In UNIX/Linux: Browse to the MicroStrategy Home folder, then to the `/bin` subfolder. Type `mstrcmdmgrw` and press **ENTER**.
2. Connect to a project source or Narrowcast Server.
3. Open the script. (From the **File** menu, select **Open**.)
4. From the **Connection** menu, select **Execute**. The script executes.

 © 2016, MicroStrategy Inc.

To execute a script from the Command Manager command line interface

For specific command syntax for the command line interface, see the Command Manager Help.

1. From the command line, type `cmdmgr.exe -interactive` and press **ENTER**. The Command Manager command line interface opens, in console mode, with an active connection-less project source connection.
2. Connect to a project source or Narrowcast Server using the `CONNECTMSTR` or `CONNECTNCS` command.
3. To load a script and execute it, type `EXECFILE` *`filename`*, where *`filename`* is the name of the script. The script is loaded into the command line interface and executed.

To invoke Command Manager from another application

Call the `cmdmgr.exe` command with the following parameters:

If the project source name, the input file, or an output file contain a space in the name or path, you must enclose the name in double quotes.

Effect	Parameters
Connection (required; choose one)	
Connect to a project source **Note**: If `-p` is omitted, Command Manager assumes a null password.	`-n` *`ProjectSourceName`* `-u` *`UserName`* `[-p` *`Password`*`]`
Initiate a connection-less project source session	`-connlessMSTR`

Connect to a Narrowcast Server
Note: If `-p` or `-s` are omitted, Command Manager assumes a null password or system prefix.
	`-w` *`ODBC_DSN`*
	`-u` *`UserName`*
	`[-p` *`Password`*`]`
	`-d` *`Database`*
	`[-s` *`SystemPrefix`*`]`

Initiate a connection-less Narrowcast Server session	`-connlessNCS`

Effect	Parameters
Note: If `-s` is omitted, Command Manager assumes a null system prefix.	`[-d Database]` `[-s SystemPrefix]`
Script input (required)	
Specify the script file to be executed **Note**: If this parameter is omitted, the Command Manager GUI is launched.	`-f InputFile`
Script output (optional; choose only one)	
Log script results, error messages, and status messages to a single file	`-o OutputFile`
Log script results, error messages, and status messages to separate files, with default file names of: • `CmdMgrResults.log` • `CmdMgrFail.log` • `CmdMgrSuccess.log`	`-break`

Log script results, error messages, and status messages to separate files, with specified names
Note: You can omit one or more of these parameters. For example, if you want to log only error messages, use only the `-of` parameter.
	`-or ResultsFile`
	`-of FailFile`
	`-os SuccessFile`

Script output options (optional)	
Begin each log file with a header containing information such as the version of Command Manager used	`-h`
Print instructions in each log file and on the console **Note**: This option is ignored if the script is encrypted. For information about encrypted Command Manager scripts, see *Encrypting Command Manager scripts, page 659*.	`-i`
If an Intelligence Server error occurred, print the Intelligence Server error code and the Command Manager exit code in each log file and on the console	`-e`
Display script output on the console	`-showoutput`
Save the results of the script in an CSV file	`-csv CSVFile`
Save the results of the script in an XML file	`-xml XMLFile`
Omit hidden objects in the script results. Hidden objects are MicroStrategy metadata objects whose HIDDEN property is set.	`-suppresshidden`
Execution options (optional)	
Halt script execution on critical errors (see *Handling execution errors, page 660*)	`-stoponerror`

A full list of parameters can also be accessed from a command prompt by entering `cmdmgr.exe -help`.

By default, the executable is installed in the following directory:

 © 2016, MicroStrategy Inc.

```
Program Files (x86)\MicroStrategy\Command Manager
```

Encrypting Command Manager scripts

By default, Command Manager scripts are saved in plain text format. This can create a security risk if your script contains a user name and password, such as for the CONNECT SERVER statement. You can avoid this security risk by saving these scripts in an encrypted format.

If you create a batch file to execute a Command Manager script from the command line, the password for the project source or Narrowcast Server login must be stored in plain text in the batch file. You can protect the security of this information by encrypting the script and having it connect to a project source or Narrowcast Server when it is executed, using the CONNECT SERVER statement. You can then execute the script from a connection-less session, which does not require a user name or password. The user name and password are provided in the Command Manager script, as part of the CONNECT SERVER statement. For detailed syntax instructions for using the CONNECT SERVER statement, see the Command Manager Help (from within the Command Manager graphical interface, press **F1**).

When you encrypt a script, you specify a password for the script. This password is required to open the script, either in the Command Manager graphical interface, or using the LOADFILE command in the Command Manager command line interface. Because a script must be opened before it can be executed in the Command Manager graphical interface, the password is required to execute the script from the graphical interface as well. However, the password is not required to execute the script from the command line or through the command line interface.

The password for an encrypted script cannot be blank, cannot contain any spaces, and is case-sensitive.

Project and configuration locking

Command Manager does not automatically lock a project or configuration when it executes statements. Thus, any time you alter a project metadata or Intelligence Server configuration with a Command Manager script, it is possible that another user could alter the metadata or configuration at the same time. This can cause metadata or configuration inconsistencies, and in the worst case may require you to reinstall Intelligence Server or restore your project from a backup.

To avoid these inconsistencies, use the LOCK PROJECT or LOCK CONFIGURATION statements in any Command Manager scripts that make changes to a project or server configuration. These statements place a lock on the metadata or configuration. A metadata lock prevents other MicroStrategy users from modifying any objects in the project in Developer or MicroStrategy Web. A configuration lock prevents other MicroStrategy users from modifying any configuration objects, such as users or groups, in the project source.

When other users attempt to open an object in a locked project or configuration, a message informs them that the project or configuration is locked because another user is

© 2016, MicroStrategy Inc.

modifying it. Users can then choose to open the object in read-only mode or view more details about the lock.

Command Manager has two kinds of locks:

- **Transient** locks are automatically released after disconnecting.
- **Permanent** locks are released only after an UNLOCK command or when the project is manually unlocked. Permanent locks are indicated by the word PERMANENT in the LOCK command.

For detailed information about locking and unlocking projects or configurations, see the *MicroStrategy Developer Help*.

If you lock a project or configuration in a Command Manager script, make sure you release the lock at the end of the script with the UNLOCK PROJECT or UNLOCK CONFIGURATION statement.

Handling syntax errors

Syntax errors occur when Command Manager encounters an instruction that it does not understand. This can be due to a typographical error (CERATE for CREATE, for example) or a statement that does not follow the required syntax in another way. For examples of the correct syntax for all Command Manager statements, see the Help.

When Command Manager encounters a syntax error, it displays the portion of the instruction set where the error was detected in the Script window and highlights the instruction. An error message is also displayed on the Messages tab of the Script window. Finally, if logging is enabled in the Options dialog box, the error message in the Messages tab is written to the log file.

Handling execution errors

Execution errors occur when an instruction is formed correctly but returns an unexpected result when it is executed. For example, attempting to delete a user who does not exist in the MicroStrategy metadata generates an execution error.

Command Manager recognizes two classes of execution errors:

- **Critical errors** occur when the main part of the instruction is not able to complete. These errors interrupt script execution when the **Stop script execution on error** option is enabled (GUI) or when the `-stoponerror` flag is used (command line).

  For example, if you submit an instruction to create a user, `user1`, that already exists in the MicroStrategy metadata database, Command Manager cannot create the user. Because creating the user is the main part of the instruction, this is a critical error. If the **Stop script execution on error** option is enabled, the script stops executing and any further instructions are ignored.

- **Noncritical errors** occur when the main part of the instruction is able to complete. These errors never interrupt script execution.

 © 2016, MicroStrategy Inc.

For example, if you submit an instruction to create a MicroStrategy user group with two members, `user1` and `user2`, but user2 does not exist in the MicroStrategy metadata database, Command Manager can still create the group. Because creating the group is the main part of the instruction (adding users is secondary), this is a noncritical error.

An error message is written to the Messages tab of the Script window for all execution errors, critical or noncritical. In addition, if logging is enabled in the Options dialog box, the error message is written to the log file.

Command Manager and prompted objects

Command Manager cannot manipulate prompted objects. For example, it cannot alter the properties of a metric that contains a prompt, and it cannot create subscriptions for a report that contains a prompt.

This restriction extends to prompts at any level of nesting. For example, if you have a custom group that contains a prompted metric, Command Manager cannot alter the properties of that custom group.

If you attempt to execute a statement that manipulates a prompted object, Command Manager returns a noncritical execution error.

Timeout errors

To avoid locking up the system indefinitely, Command Manager has a built-in timeout limit of 20 minutes. If a statement has been executing for 20 minutes with no response from Intelligence Server, Command Manager reports a request timeout error for that command and executes the next instruction in the script. However, Command Manager does not attempt to abort the command. In some cases, such as database-intensive tasks such as purging the statistics database, the task may continue to execute even after Command Manager reports a timeout error.

The following statements are not subject to the 20-minute Command Manager timeout limit. A script containing these statements continues executing until Intelligence Server reports that the task has succeeded or failed.

- Create Project statement
- Update Project statement
- Update Privileges statement
- Import Package statement

© 2016, MicroStrategy Inc.

Command Manager script syntax

The Command Manager script engine uses a unique syntax that is similar to SQL and other such scripting languages. For a complete guide to the commands and statements used in Command Manager, see the Command Manager Help.

A Command Manager script consists of one or more script statements. Each statement ends with a semicolon (`;`).

Statements consist of one or more tokens. A token is a word, a list of words enclosed in quotation marks, or a symbol. A token is recognized by Command Manager as an individual unit with a specific meaning. Tokens can be:

- **reserved words**, which are words with a specific meaning in a Command Manager script. For a complete list of reserved words, see the Command Manager Help.
- **identifiers**, which are words that the user provides as parameters for the script. For example, in the statement `LIST MEMBERS FOR USER GROUP "Managers";` the word `Managers` is an identifier. Identifiers must be enclosed in quotation marks.

  In general, either double quotes or single quotes can be used to enclose identifiers. However, if you want to include either single quotes or double quotes as part of an identifier, you must either enclose that identifier in the other kind of quotes, or put a caret in front of the interior quote. For example, to refer to a metric named `Count of "Outstanding" Customer Ratings`, you would need to use one of the following methods:

  — Use single quotes to enclose the identifier:

  ```
  'Count of "Outstanding" Customer Ratings'
  ```

  — Use double quotes to enclose the identifier and put carets in front of the interior double quotes:

  ```
  "Count of ^"Outstanding^" Customer Ratings"
  ```

  If your identifier contains double-byte characters, such as characters used in the Korean, Japanese, or Chinese character sets, you must enclose the identifier in square brackets `[ ]`. If the identifier is also enclosed in quotation marks, these square brackets must be placed inside the quotation marks.

- **symbols**, such as `; , ' " ^`

  The caret (`^`) functions as an escape character. It causes any other special character that follows it to be treated literally and not interpreted as a special character. If you want to include a literal caret in your statement, you must precede it with another caret. For example, if you have a user group named `^Control`, in Command manager scripts you must refer to it as `^^Control`.

- **numbers** in any notation

© 2016, MicroStrategy Inc.

- **dates**
- **object GUIDs**
- **other special characters** such as carriage returns, tabs, or spaces

Using Command Manager from the command line

In addition to the graphical user interface and the command line execution, Command Manager has a text-based command line interface. With this interface, you can create and execute Command Manager scripts in an environment where the graphical user interface is unavailable, such as when accessing a UNIX system via telnet.

When you start the command line interface, it is in console mode, with a connection-less project source connection. The command prompt in console mode displays the metadata source and user to which Command Manager is connected.

Entering a Command Manager script instruction switches Command Manager into edit mode. From edit mode you can continue typing your script. You can also save or execute the script.

To see a list of instructions for the command line interface, from the command line interface type `help` and press **ENTER**. A list of Command Manager command line instructions and an explanation of their effects is displayed.

To start the Command Manager command line interface

From the command line, type `cmdmgr.exe -interactive` and press **ENTER**. The Command Manager command line interface opens, in console mode, with an active connection-less project source connection.

Using Command Manager with OEM software

Developers of Original Equipment Manufacturer (OEM) applications that use embedded MicroStrategy projects may find that they need flexibility in configuring their environment. Command Manager Runtime is a slimmed-down version of the Command Manager command-line executable for use with these OEM applications. For information about obtaining Command Manager Runtime, contact your MicroStrategy sales representative.

© 2016, MicroStrategy Inc.

Command Manager Runtime uses a subset of the commands available for the full version of Command Manager. If you try to execute a script with statements that are not available in Command Manager Runtime, the script fails with the message, “You are not licensed to run this command.” For a list of the commands available in Command Manager Runtime, with syntax and examples for each command, see the *Command Manager Runtime* chapter in the *Supplemental Reference for System Administration*.

 © 2016, MicroStrategy Inc.

16

VERIFYING REPORTS AND DOCUMENTS WITH INTEGRITY MANAGER

Introduction

MicroStrategy Integrity Manager is an automated comparison tool designed to streamline the testing of MicroStrategy reports and documents in projects. This tool can determine how specific changes in a project environment, such as the regular maintenance changes to metadata objects or hardware and software upgrades, affect the reports and documents in that project.

For instance, you may wish to ensure that the changes involved in moving your project from a development environment into production do not alter any of your reports. Integrity Manager can compare reports in the development and the production projects, and highlight any differences. This can assist you in tracking down discrepancies between the two projects.

You can use Integrity Manager to execute reports or documents from a single MicroStrategy project to confirm that they remain operational after changes to the system. Integrity Manager can execute any or all reports from the project, note whether or not those reports execute, and show you the results of each report.

Integrity Manager can also test the performance of an Intelligence Server by recording how long it takes to execute a given report or document. You can execute the reports or documents multiple times in the same test and record the time for each execution cycle, to get a better idea of the average Intelligence Server performance time. For more

© 2016, MicroStrategy Inc.

information about performance tests, see *Testing Intelligence Server performance, page 668*.

For reports you can test and compare the SQL, grid data, graph, Excel, or PDF output. For documents you can test and compare the Excel or PDF output, or test whether or not the documents execute properly. If you choose not to test and compare the Excel or PDF output, no output is generated for the documents. Integrity Manager still reports whether or not the documents executed successfully and how long it took them to execute.

Note the following:

- To execute an integrity test on a project, you must have the Use Integrity Manager privilege for that project.
- Integrity Manager can only test projects in Server (three-tier) mode. Projects in Direct Connection (two-tier) mode cannot be tested with this tool.
- To test the Excel export of a report or document, you must have Microsoft Excel installed on the machine running Integrity Manager.

This section describes how to use Integrity Manager to view and compare reports and documents. Topics include:

- *What is an integrity test?, page 666*
- *Best practices for using Integrity Manager, page 669*
- *Creating an integrity test, page 670*
- *Executing an integrity test, page 672*
- *Viewing the results of a test, page 683*
- *List of tags in the integrity test file, page 688*

What is an integrity test?

In an *integrity test*, Integrity Manager executes reports or documents from a base project and informs you as to which reports and documents failed to execute. Depending on the type of integrity test, Integrity Manager may compare those reports and documents against those from another project, or from a previously established baseline. An integrity test may also involve comparing reports and/or documents from two previously established baselines, and not executing against an Intelligence Server at all.

The Integrity Manager Wizard walks you through the process of setting up integrity tests. You specify what kind of integrity test to run, what reports or documents to test, and the execution and output settings. Then you can execute the test immediately, or save the test for later use and re-use. For information on reusing tests, see *Saving and loading a test, page 671*.

© 2016, MicroStrategy Inc.

Types of integrity tests

A **single-project** integrity test confirms that reports and documents from a project execute to completion, without errors. This is useful when changes have been made somewhere in the system, and you want to ensure that none of the changes cause execution errors in your reports or documents.

In a single-project test, Integrity Manager executes the specified reports and documents. It then displays a list of the reports along with whether the execution of each report or document succeeded or failed. If a report or document failed, you can double-click on the report name in the results list to see what error message was generated.

In addition to the single-project integrity test, Integrity Manager supports these types of comparative integrity tests:

- **Project-versus-project** integrity tests compare reports and/or documents from two different projects. This is useful when you are moving a project from one environment to another (for instance, out of development and into production), and you want to ensure that the migration does not cause changes in any reports or documents in the project.
- **Baseline-versus-project** integrity tests compare reports and/or documents from a project against a previously established baseline. The baseline can be established by running a single-project integrity test, or taken from a previous execution of a project-versus-project integrity test.

  Baseline-versus-project tests can be used as an alternative to project-versus-project tests when no base project is available, or when running against a production Intelligence Server would be too costly in terms of system resources. Also, by using baseline-versus-project tests a user can manually change the results which he wants to compare the target project with.
- **Baseline-versus-baseline** integrity tests compare reports and/or documents from two previously established baselines against each other. These baselines can be established by running single-project integrity tests (see below), or taken from a previous execution of a project-versus-project integrity test.

  These tests can be useful if you have existing baselines from previous tests that you want to compare. For example, your system is configured in the recommended project life cycle of development -> test -> production (for more information on this life cycle, see the *Managing your projects* chapter in the *System Administration Guide*). You have an existing baseline from a single project test of the production project, and the results of a project versus project test on the development and test projects. In this situation, you can use a baseline versus baseline test to compare the production project to the test project.

In each of these comparative tests, Integrity Manager executes the specified reports and documents in both the baseline and the target. You can compare the report data, generated SQL code, graphs, Excel exports, and PDF output for the tested reports; you can compare the Excel exports and PDF output for tested documents, or test the execution of the documents without exporting the output. Integrity Manager informs you which reports and documents are different between the two projects, and highlights in red the differences between them.

© 2016, MicroStrategy Inc.

Testing Intelligence Server performance

In addition to testing reports and documents for execution and for accuracy between projects, Integrity Manager can determine how long it takes an Intelligence Server to execute a given set of reports or documents. This is called a performance test. You can execute the reports and documents in the integrity test multiple times, to get a better idea of the average time it takes to execute each report.

In a performance test, Integrity Manager records the time it takes to execute each report or document. If the reports and documents are being executed more than once, Integrity Manager records each execution time. You can view the minimum, maximum, and average execution time for each report or document in the Results Summary area. In a comparative integrity test, you can also view the difference in time between the baseline and target reports and documents.

Performance test best practices

The results of a performance test can be affected by many factors. The following best practices can help ensure that you get the most accurate results from a performance test:

- Performance comparison tests should be run as single-project integrity tests. This reduces the load on Integrity Manager and ensures that the recorded times are as accurate as possible.

  To compare performance on two Intelligence Servers, MicroStrategy recommends following the steps below:

  a Perform a single project test against one project, saving the performance results.

  b Perform a single project test against the second project, saving the performance results.

  c Compare the two performance results in a baseline-versus-baseline test.

- Wait until the performance test is complete before attempting to view the results of the test in Integrity Manager. Otherwise the increased load on the Integrity Manager machine may cause the recorded times to be increased for reasons not related to Intelligence Server performance.

- If you are using a baseline-versus-project test or a baseline-versus-baseline test, make sure that the tests have processed the reports and/or documents in the same formats. Execution times are not recorded for each format, only for the aggregate generation of the selected formats. Thus, comparing a baseline with SQL and Graph data against a test of only SQL data is likely to give inaccurate results.

- If the **Use Report Cache** setting is selected on the Select Execution Settings page of the Integrity Manager Wizard, make sure that a valid cache exists for each report to be tested. Otherwise the first execution cycle of each report takes longer than the subsequent cycles, because it must generate the cache for the other cycles to use. One way to ensure that a cache exists for each report is to run a single-project integrity test of each report before you run the performance test.

This setting only affects reports, and does not apply to documents.

 © 2016, MicroStrategy Inc.

- In the Integrity Manager wizard, on the Select Execution Settings page, make sure **Concurrent Jobs** is set to **1**. This causes Intelligence Server to run only one report or document at a time, and provides the most accurate benchmark results for that Intelligence Server.
- The **Cycles** setting on the Select Processing Options page of the Integrity Manager Wizard indicates how many times each report or document is executed. A high value for this setting can dramatically increase the execution time of your test, particularly if you are running many reports or documents, or several large reports and documents.

Best practices for using Integrity Manager

MicroStrategy recommends the following best practices when using Integrity Manager:

- Run large integrity tests during off-peak hours, so that the load on Intelligence Server from the integrity test does not interfere with normal operation. You can execute integrity tests from the command line using a scheduler, such as the Windows AT scheduler. For information about executing integrity tests from the command line, see *Executing a test from the command line, page 673*.
- Before performing a system upgrade, such as a database upgrade or a MicroStrategy metadata upgrade, create a baseline of the reports you want to test. You can create this baseline by executing a single-project integrity test. Then, after the upgrade, you can verify the upgrade process by executing a baseline-versus-project test of the baseline and the upgrade project.
- Understand how Integrity Manager answers prompted reports, and how you can configure the answers to prompted reports, as described in *Executing prompted reports with Integrity Manager, page 676*.
    - If you are having trouble comparing prompted reports, you can save static versions of those reports in a “regression test” folder in each project, and use those static reports for integrity tests.
- If your MicroStrategy security configuration involves security filters, make sure that the user executing the integrity test has the same security filters for both projects. For example, you can create a test user who has the same security filter for each project, and execute all integrity tests under this user.
    - Alternately, you can execute the test using multiple MicroStrategy users, as described in *Executing a test under multiple MicroStrategy user accounts, page 680*. Make sure that the users that you are comparing have matching security filters. For example, if User1 is assigned security filter FilterA in project Project1, make sure you compare the reports with a user who is also assigned security filter FilterA in project Project2.
- When you are comparing graph reports and noting the differences between the graphs, adjust the Granularity slider so that the differences are grouped in a way that is useful. For more information about how Integrity Manager evaluates and groups

© 2016, MicroStrategy Inc.

differences in graph and PDF reports, see *Grouping differences in graph and PDF reports, page 685*.

- If you are executing a performance test, follow the best practices described in *Performance test best practices, page 668*.

Creating an integrity test

The following high-level procedure provides an overview of the steps involved in creating an integrity test. For an explanation of the information required at any given page in the wizard, see the Help (from the wizard, click **Help**, or press **F1**).

To create an integrity test

1. Start Integrity Manager. (From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Products**, and then select **Integrity Manager**.)
2. From the **File** menu, select **Create Test**. The Integrity Manager Wizard opens and the Welcome page is displayed.
3. Select the type of test you want to create:
   - To compare reports and documents from two projects, select **Project versus project**.
   - To compare reports and documents against a previously established baseline, select **Baseline versus project**.
   - To compare reports and documents from two previously established baselines, select **Baseline versus baseline**.
   - To confirm that reports and documents in a project execute without errors, select **Single project**.
4. Specify the baselines and projects to be tested. For each project, provide a MicroStrategy login and password with the Use Integrity Manager privilege for that project.
5. Select the reports and/or documents to be tested. You can select individual reports or documents, or entire folders. You can also select search objects; in this case, Integrity Manager tests all reports and documents from the results of the search object.

If you select any Intelligent Cube reports, make sure that the Intelligent Cube the reports are based on has been published before you perform the integrity test. Integrity Manager can test the SQL of Intelligent Cubes even if they have not

 © 2016, MicroStrategy Inc.

been published, but cannot test Intelligent Cube reports based on an unpublished Intelligent Cube.

6 Specify test execution options, such as how to answer any unanswered prompts, what details to log, and whether to use report caches.

7 Select what types of analysis to perform. For reports, you can analyze any or all of the grid data, underlying SQL, graph data, Excel export, or PDF output. For documents you can analyze the Excel export or PDF output.

Only reports that have been saved in Graph or Grid/Graph view can be analyzed as graphs.

You can also select to record the execution time of each report and/or document, to analyze the performance of Intelligence Server.

8 Review the information presented on the Summary page.

9 To save the settings for use in later tests, click **Save Test**. Navigate to the desired directory, enter a file name, and click **OK**.

For instructions on executing a saved test, see *Saving and loading a test, page 671*.

10 To execute the test immediately, regardless of whether you saved the settings, click **Run**. The Integrity Manager Wizard closes and Integrity Manager begins to execute the selected reports and documents. As the reports execute, the results of each report or document appear in the Results Summary area of the Integrity Manager interface.

Saving and loading a test

When you have set up a test using the Integrity Manager Wizard, you can choose to save your settings to a file. This enables you to execute the same test at a later time without having to re-create the test. For example, a metric currently being developed is causing errors in several reports. You can execute a test on those reports to check whether the metric still causes execution errors. Saving the test settings makes it easy to run this test once the latest version of the metric is ready.

For security reasons, the passwords for the project logins (provided on the Enter Base Project Information page and Enter Target Project Information page) are not saved to the test file. You must re-enter these passwords when you load the test.

To save test settings

1 Step through the Integrity Manager Wizard and answer its questions. For detailed instructions, see *Creating an integrity test, page 670*.

2 When you reach the Summary page of the Integrity Manager Wizard, click **Save Test**. A Save dialog opens.

3 Navigate to the desired folder and enter a file name to save the test as. By default this file will have an extension of `.mtc`.

4 Click **OK**. The test settings are saved to the specified file.

You can execute the test immediately by clicking **Run**. The Integrity Manager Wizard closes and Integrity Manager begins to execute the selected reports and documents. As they execute, their results appear in the Results Summary area of the Integrity Manager interface.

To load a previously saved test

1 In Integrity Manager, from the **File** menu select **Load Test**. An Open File dialog box opens.

2 Navigate to the file containing your test information and open it. The Integrity Manager Wizard opens at the Welcome page. The settings for the test are loaded into each page of the wizard.

The default extension for integrity test files is `.mtc`.

3 Step through the wizard and confirm the settings for the test.

4 At the **Enter Base Project Information** page and **Enter Target Project Information** page, enter the password for the login used to access the base or target project.

5 When you reach the Summary page, review the information presented there. When you are satisfied that the test settings shown are correct, click **Run**. The Integrity Manager wizard closes and Integrity Manager begins to execute the selected reports and documents. As they execute, their results appear in the Results Summary area of the Integrity Manager interface.

Executing an integrity test

After creating or loading an integrity test, you can execute it by clicking **Run** from the Summary page of the Integrity Manager wizard. You can also execute a saved test from the command line, without launching the Integrity Manager graphical interface. For instructions, see *Executing a test from the command line, page 673*.

You can also re-run reports in a test that has just finished execution. For example, a number of reports in an integrity test may fail because of an error in a metric. You can correct the metric and then re-run those reports to confirm that the reports now match. To re-run the reports, select them, and then from the **Run** menu, select **Refresh selected items**.

 © 2016, MicroStrategy Inc.

Executing a test from the command line

Integrity Manager's command line interface enables you to execute a test without having to load the graphical interface, or to schedule a test to run at specific times or dates. For example, you perform routine maintenance on your data warehouse every month. Using the Windows `AT` command or the UNIX scheduler, you can schedule a baseline-versus-project test to run every month immediately after routine database maintenance. This ensures that no reports are broken during maintenance.

If you are running Integrity Manager in a Windows environment, you must be logged in to Windows with an Administrator account. In addition, if you are running Integrity Manager directly from the command prompt, you must set the command prompt to run with full administrative privileges. To do this, right-click on the command prompt shortcut and select **Run As**. Clear the **Run this program with restricted access** check box and click OK.

To execute a previously saved integrity test from the command line

After creating and saving a test (for instructions, see *To save test settings, page 671*), call the Integrity Manager executable `MIntMgr.exe` with the parameters listed in the table below. All parameters are optional except the -f parameter, which specifies the integrity test file path and name.

By default, the executable is installed in the following directory:

```
C:\Program Files (x86)\MicroStrategy\Integrity Manager
```

Effect	Parameters
Integrity test file path and name	`-f FileName`
Base system password. **Note**: For instructions on how to specify multiple passwords, or passwords using special characters, see *Password syntax, page 675*.	`-b BasePassword`
Target system password. **Note**: For instructions on how to specify multiple passwords, or passwords using special characters, see *Password syntax, page 675*.	`-t TargetPassword`
Note: The following parameters modify the execution of the test. They do not modify the `.mtc` test file.	
Output directory. **Note**: This directory must exist before the test can be executed.	`-o OutputDirectory`
Log file path and name.	`-logfile LogfileName`
Base server name.	`-bserver BaseServer`

Effect	Parameters
Target server name.	`-tserver TargetServer`
Base server port number.	`-bport BasePort`
Target server port number.	`-tport TargetPort`
Base project.	`-bproject BaseProject`
Target project.	`-tproject TargetProject`
Login for base project. **Note**: For multiple logins, enclose all logins in double quotes (“) and separate each login with a comma (,).	`-blogin BaseLogin` `-blogin “BaseLogin1, .., BaseLoginN”`
Login for target project. **Note**: For multiple logins, enclose all logins in double quotes (“) and separate each login with a comma (,)	`-tlogin TargetLogin` `-tlogin “TargetLogin1, .., TargetLoginN”`
Base baseline file path and name. **Note**: The GUIDs of objects to be tested in the baseline file must match any GUIDs specified in the `.mtc` file.	`-bbaselinefile BaseBaselineFile`
Target baseline file path and name. **Note**: The GUIDs of objects to be tested in the baseline file must match any GUIDs specified in the `.mtc` file.	`-tbaselinefile TargetBaselineFile`
Whether to use load balancing in the base project, that is, whether to execute the reports and documents across all nodes of the cluster (`True`) or on a single node (`False`). **Note**: If this option is used, it overrides the setting in the integrity test file.	`-bloadbalance true` `-bloadbalance false`
Whether to use load balancing in the target project, that is, whether to execute the reports and documents across all nodes of the cluster (`True`) or on a single node (`False`). **Note**: If this option is used, it overrides the setting in the integrity test file.	`-tloadbalance true` `-tloadbalance false`
GUID of the test folder. If this option is used, the reports and documents specified in the integrity test file are ignored. Instead, Integrity Manager executes all reports and documents in the specified folder. **Note**: This option can only be used with a single-project integrity test or a project-versus-project integrity test.	`-folderid FolderGUID`

 © 2016, MicroStrategy Inc.

Password syntax

When specifying passwords with special characters, or specifying multiple passwords, use the following syntax:

- If a password contains a single quote (') or a comma (,), that character must be preceded by a single quote.
- If a password contains a double quote (") that character must be substituted by ". If a password contains an ampersand (&) that character must be substituted by &.

  For example, if the password is `12'&ABC"12,3` then the password must be specified as `12''&ABC"12',3`.
- If multiple logins are used, a password must be specified for each login. The entire list of passwords must be enclosed in double quotes (") and the passwords must be separated by a comma (,).
- If multiple passwords are used and a user in the base project or target project has an empty password, the position of that user's password in the list of passwords is indicated by a space between commas.

  For example, if the users for an integrity test are User1, User2, and User3, and User2 has an empty password, the list of passwords is `"password1, ,password3"`.

Command line exit codes

When an integrity test that has been executed from the command line ends, it returns a number. This number is an exit code. If the script ends unexpectedly, this exit code can help you find the cause of the error.

To view the error code, in the same command prompt window as the test execution, type `echo %ERRORLEVEL%` and press Enter.

Exit code	Meaning
0	The test execution succeeded and all reports have a status of Matched.
1	The test execution succeeded, but at least one report has a status other than Matched.
2	Integrity Manager was unable to establish a connection to Intelligence Server, or the connection was interrupted during the test.
3	Either your Integrity Manager license has expired, or you do not have the privileges necessary to run Integrity Manager. You can view license information in License Manager.
4	The test execution failed. For more information about this error, see the integrity test log for this test.
5	The test execution was aborted by the user.

Manually editing an integrity test

If you need to make minor changes to an integrity test, it may be faster to make those changes by editing the test file, rather than stepping through the Integrity Manager Wizard.

The test file is a plain-text XML file, and can be edited in a text editor, such as Notepad. For an explanation of all the XML tags included in the test file, see *List of tags in the integrity test file, page 688*.

Executing a test against a remote Intelligence Server

Integrity Manager uses the Windows TCP/IP `hosts` file to contact remote Intelligence Servers. This file contains server names and IP addresses for other networked machines that can be accessed from this machine.

In Windows, to execute an integrity test against an Intelligence Server on a machine other than the one Integrity Manager is running on, you need to add an entry to the `hosts` file for the machine Integrity Manager is running on.

To add an entry to the hosts file

1. In the Windows system folder, navigate to the `\system32\drivers\etc` folder.
   - For Windows 2000 and Windows 2003, the system folder is `C:\WINNT\`.
   - For Windows XP, the system folder is `C:\WINDOWS\`.
2. Open the `hosts` file with a text editor, such as Notepad.
3. For each Intelligence Server machine that you want to test against, add a line to the file in the same format as the examples given in the file.
4. Save and close the `hosts` file. You can now execute integrity tests against the Intelligence Servers specified in the file.

Executing prompted reports with Integrity Manager

In a prompted report, the user specifies certain objects, such as the elements of an attribute, or the range of values for a metric. For an introduction to prompts, see the *MicroStrategy Basic Reporting Guide*.

Integrity Manager can use any of the following methods to resolve prompts:

- **Personal answer**: Personal answers are default prompt answers that are saved for individual MicroStrategy logins. Any prompts with personal answers saved for the login using Integrity Manager can be resolved using those personal answers.
- **Default object answer**: A prompted report can have two possible default answers: a default answer saved with the prompt, and a default answer saved with the report.

 © 2016, MicroStrategy Inc.

These default answers can be used to resolve the prompt. If both default answers exist, Integrity Manager uses the answer saved with the report.

- **Integrity Manager user-defined answer**: Any required value and hierarchy prompts can be answered according to the defaults provided in the Select Prompt Settings page. You can provide default answers for value prompts, and a default number of elements for hierarchy prompts.
- **Integrity Manager internal answer**: Integrity Manager can use its internal logic to attempt to answer any other required prompts without default answers. For example, a prompt that requires a certain number of elements to be selected from a list can be answered by selecting the minimum number of elements from the beginning of the list.

By default Integrity Manager uses all of these options, in the order listed above. You can disable some options or change the order of the options in the Advanced Options dialog box in the Integrity Manager Wizard.

For example, you may want to never use your personal answers to answer prompts, and use the user-defined answers instead of the default answers for value prompts. You can configure the user-defined answers for value prompts in the Select Prompt Settings page. Then, in the Advanced Options dialog box, clear the **Personal answer** check box and move **Integrity Manager user-defined answer** above **Default object answer**.

Optional prompts

You control whether Integrity Manager answers optional prompts on the Select Prompt Settings page of the Integrity Manager Wizard.

- To answer optional prompts in the same way as required prompts, select the **Answer optional prompts** check box.
- To leave optional prompts that do not have default or personal answers unanswered, clear the **Answer optional prompts** check box.

Using non-default personal answers in prompts

By default, when Integrity Manager answers a prompt with a personal answer, it uses only the default personal answer for each prompt. If a prompt does not have a default personal answer for the current user, Integrity Manager moves to the next method of prompt resolution.

To change this default, in the Advanced Options dialog box, select the **Group personal prompt answers by their names** option. When this option is selected, Integrity Manager executes each report/document once for each personal answer for each prompt in the report/document. If multiple prompts in the report/document have personal answers with the same name, those personal answers are used for each prompt in a single execution of the report/document.

For personal prompt answers to be grouped, the answers must have the exact same name. For example, if the base project contains a personal prompt answer named

`AnswerA` and the target project contains a personal prompt answer named `Answer_A`, those prompt answers will not be grouped together.

For example, consider a report with two prompts, Prompt1 and Prompt2. The user executing the report has personal answers for each of these prompts. The personal answers are named as follows:

Prompt	Answers
Prompt1	AnswerA, AnswerB
Prompt2	AnswerA, AnswerC, AnswerD

Integrity Manager executes this report four times, as shown in the table below:

Execution	Prompt 1 answer	Prompt 2 answer
1	Personal answer AnswerA	Personal answer AnswerA
2	Personal answer AnswerB	(next prompt answer method)
3	(next prompt answer method)	Personal answer AnswerC
4	(next prompt answer method)	Personal answer AnswerD

Since Prompt1 and Prompt2 both have a personal answer saved with the name AnswerA, Integrity Manager groups those answers together in a single execution. Only Prompt1 has an answer named AnswerB, so Integrity Manager executes the report with AnswerB for Prompt1 and uses the next available method for answering prompts to answer Prompt2. In the same way, only Prompt2 has answers named AnswerC and AnswerD, so when Integrity Manager executes the report using those answers for Prompt2 it uses the next available prompt answer method for Prompt1.

Unanswered prompts

If a prompt cannot be answered by Integrity Manager, the report execution fails and the report's status changes to **Not Supported**. A detailed description of the prompt that could not be answered can be found in the Details tab of the Report Data area for that failed report. To view this description, select the report in the Results summary area and then click the Details tab.

You can configure Integrity Manager to open a Not Supported report in MicroStrategy Web. You can answer any prompts manually and save the report. Integrity Manager then executes the newly saved report, using the specified prompt answers.

Prompts that cannot be answered by Integrity Manager's internal logic include:

- Prompts that cannot be answered at all, such as an element list prompt that contains no elements in the list
- Level prompts that use the results of a search object to generate a list of possible levels

 © 2016, MicroStrategy Inc.

- Prompted metric qualifications (used in filters or custom groups)
- MDX expression prompts

To resolve unanswered prompts in MicroStrategy Web

Configure the integrity test to open the reports in MicroStrategy Web

1. Create an integrity test. Step through the Integrity Manager Wizard and enter the information required on each page. For details about the information required on each page, click **Help** to open the help for that page of the wizard.
2. In the Select Prompt Settings page, click **Advanced Options**. The Advanced Options dialog box opens.
3. Select the **Link to MicroStrategy Web for unresolved prompts** check box.
4. In the **URL for Base connection** and **URL for Target Connection** fields, type the URL for the baseline and target projects' Web servers. To test each URL, click the Test button. If it is correct, a browser window opens at the main MicroStrategy Web page for that server.

   The default URL for MicroStrategy Web is:
   `http://webservername/MicroStrategy/asp/Main.aspx`

   and the default URL for Web Universal is:
   `http://webservername:8080/MicroStrategy/servlet/mstrWeb`

   where *webservername* is the name of your MicroStrategy Web server machine.
5. Click **OK**. The Advanced Options dialog box closes.
6. Finish defining the test, then execute it.

Resolve the prompts in MicroStrategy Web

7. If any reports contain prompts that cannot be resolved by Integrity Manager, the Link to MicroStrategy Web for Unresolved Prompts dialog box opens.
8. To save the report with the correct prompt answers, click the report's name in the dialog box. The report opens in MicroStrategy Web, in a new browser window.

   Note: If a Login dialog box opens, select an authentication method, enter a username and password, and click **OK**.
9. Answer the prompts for the report and save it. Depending on your choices in the Advanced Options dialog box, you may need to save the report as a static, unprompted report.

10 In Integrity Manager, click **Continue**. Integrity Manager executes the newly saved version of the report.

To continue the integrity test without re-running the report, click **Ignore**. The report is listed in the Results Summary area with a status of **Not Supported**. To skip all future requests to resolve prompts in MicroStrategy Web for this integrity test, click **Ignore All**.

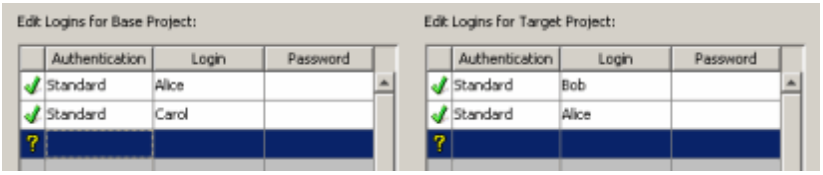
Executing a test under multiple MicroStrategy user accounts

When you create an integrity test, you can specify multiple MicroStrategy user accounts to execute the reports and documents in the test.

For example, your MicroStrategy system may use security filters to restrict access to data for different users. If you know the MicroStrategy login and password for a user who has each security filter, you can run the integrity test under each of these users to ensure that the security filters are working as designed after an upgrade. You can also compare a set of reports from the same project under two different users to ensure that the users are seeing the same data.

On the Enable Multiple Logins page of the Integrity Manager Wizard, you specify the authentication method, MicroStrategy login, and password for each user. Integrity Manager executes each report/document in the integrity test under each user account, one account at a time, in the order the accounts are listed. If you are executing a comparative integrity test, the results from the first user in the base project are compared with the results from the first user in the target project, and so on.

For example, you create a project-versus-project integrity test with reports Report1, Report2, and Report3. You are testing the reports with users Alice and Carol in the base project. You want to compare Alice's results in the base project with Bob's results in the target project, and Carol's results in the base project with Alice's results in the target project, so you configure the Enable Multiple Logins page as follows:



When the test is executed, the reports are executed in the following order:

Report execution	Base project report and user	Target project report and user
1	Report1 Alice	Report1 Bob
2	Report2 Alice	Report2 Bob
3	Report3 Alice	Report3 Bob

 © 2016, MicroStrategy Inc.

Report execution	Base project report and user	Target project report and user
4	Report1 Carol	Report1 Alice
5	Report2 Carol	Report2 Alice
6	Report3 Carol	Report3 Alice

Note that the reports executed by Alice in the base project are compared with the reports executed by Bob in the target project, and the reports executed by Carol in the base project are compared with the reports executed by Alice in the target project.

To execute a test with multiple users

1. Create an integrity test, including the information described in the steps below. Step through the Integrity Manager Wizard and enter the information required on each page. For details about the information required on each page, click **Help** to open the help for that page of the wizard.
2. On the Welcome page, select the **Enable Multiple Logins** check box.
3. On the Enable Multiple Logins page, for each user, specify the authentication mode, login, and password.
4. Make sure the users are in the order that you want the test to be executed in. In addition, if you are creating a comparative integrity test, make sure that the users whose results you want to compare are paired up correctly in the tables.
5. Finish stepping through the wizard and entering the required information. When the test is executed, each report/document is executed under each specified user account.

Ignoring dynamic SQL when comparing SQL

Dynamic SQL generates SQL statements that are partially created at the time of execution. Dynamic SQL may be generated differently in the base project and in the target project, so it can cause reports to be flagged as **Not Matched** even if the report SQL is otherwise identical.

You can configure Integrity Manager to ignore dynamic SQL in its comparison. To do this, make changes in two places: a report's VLDB properties and in Integrity Manager.

To configure Integrity Manager to ignore dynamic SQL

1. For reports that use dynamic SQL, enclose the dynamic SQL in identifying SQL comments. Enter the comments in the VLDB properties Pre/Post statements. For

information on these statements, dynamic SQL syntax and parameters, and for steps on changing VLDB properties, see the *Supplemental Reference for System Administration*.

For example, before each section of dynamic SQL, include a beginning comment line, such as:

```
/* BEGIN DYNAMIC SQL */
```

At the end of each section of dynamic SQL, include an ending comment line, such as:

```
/* END DYNAMIC SQL */
```

2. In Integrity Manager, create a comparative integrity test by stepping through the Integrity Manager wizard.
3. On the Select Processing Options page, select the **SQL/MDX** check box, then click **Advanced Options**.
4. Select the **SQL/MDX** category.
5. In the **Dynamic SQL Start** field, type the text that matches the text you entered in the VLDB properties to indicate the beginning of the dynamic SQL. For this example, type `/* BEGIN DYNAMIC SQL */`
6. In the **End** field, type the text that matches the text you entered in the VLDB properties to indicate the end of the dynamic SQL. For this example, type `/* END DYNAMIC SQL */`
7. Click **OK**, then continue through the wizard.

Matching equivalent SQL strings

Sometimes reports in the base project and the target project include SQL that is functionally equivalent but slightly different. For example, reports in the base project might use a table prefix of `TEST` while reports in the target project use a table prefix of `PROD`. You want Integrity Manager to treat the table prefixes as identical for purposes of comparison, because reports that differ only in their table prefixes should be considered identical.

In this case, you can use the SQL Replacement feature to replace `TEST` with `PREFIX` in the base project, and `PROD` with `PREFIX` in the target project. Now, when Integrity Manager compares the report SQL, it treats all occurrences of `TEST` in the base and `PROD` in the target as `PREFIX`, so they are not considered to be differences.

The changes made by the SQL Replacement Table are not stored in the SQL files for each report. Rather, Integrity Manager stores those changes in memory when it executes the integrity test.

Access the SQL Replacement feature from the Advanced Options dialog box, on the Select Processing Options page of the Integrity Manager wizard.

 © 2016, MicroStrategy Inc.

Viewing the results of a test

Once you have started executing a test, information about the reports and documents being tested appears in the Results Summary area of Integrity Manager. This area lists all the selected reports and documents, by name and path. Each report or document also shows one of the following statuses:

- **Pending** reports and documents have not yet begun to execute.
- **Running** reports and documents are in the process of executing.

  In a performance test, this status appears as **Running (#/#)**. The first number is the current execution cycle. The second number is the number of times the report or document will be executed in the test.
- **Paused (#/#)** reports and documents, in a performance test, have executed some but not all of their specified number of cycles when the test execution is paused. The first number is the number of cycles that have been executed. The second number is the number of times the report or document will be executed in the test.
- **Completed** reports and documents have finished their execution without errors.
- **Timed Out** reports and documents did not finish executing in the time specified in the **Max Timeout** field in the Select Execution Settings page. These reports and documents have been canceled by Integrity Manager and will not be executed again during this run of the test.
- **Error** indicates that an error has prevented this report or document from executing correctly. To view the error, double-click the status. The report details open in the Report Data area of Integrity Manager, below the Results Summary area. The error message is listed in the Execution Details section.
- **Not Supported** reports and documents contain one or more prompts for which an answer could not be automatically generated. To see a description of the errors, double-click the status. For details of how Integrity Manager answers prompts, see *Executing prompted reports with Integrity Manager, page 676*.

Additional information for **Completed** reports and documents is available in the **Data**, **SQL**, **Graph**, and **Excel** columns:

- **Matched** indicates that the results from the two projects are identical for the report or document. In a single-project integrity test, **Matched** indicates that the reports and documents executed successfully.
- **Not Matched** indicates that a discrepancy exists between the two projects for the report or document. To view the reports or documents from each project in the Report Data area, select them in the Results Summary area.
- **Not Compared** indicates that Integrity Manager was unable to compare the reports and documents for this type of analysis. This can be because the report or document was not found in the target project, because one or more prompts are not supported

© 2016, MicroStrategy Inc.

by Integrity Manager, or because an error prevented the report or document from executing.

- **Not Available** N/A indicates that Integrity Manager did not attempt to execute the report or document for this type of analysis. This may be because this type of analysis was not selected on the Select Processing Options page, or (if N/A is present in the **Graph** column) because the report was not saved as a Graph or Grid/Graph.

To view a **Completed** report or document and identify discrepancies, select its entry in the Results Summary. The report or document appears in the Report Data area of Integrity Manager, below the Results Summary.

In a comparative integrity test, both the base and the target report or document are shown in the Report Data area. Any differences between the base and target are highlighted in red, as follows:

- In the Data, SQL, or Excel view, the differences are printed in red. In Data and Excel view, to highlight and bold the next or previous difference, click the **Next Difference** or **Previous Difference** icon.
- In the Graph view, the current difference is circled in red. To circle the next or previous difference, click the **Next Difference** or **Previous Difference** icon. To change the way differences are grouped, use the Granularity slider. For more information about differences in graph reports, see *Grouping differences in graph and PDF reports, page 685*.

Viewing graphs in Overlap layout enables you to switch quickly between the base and target graphs. This layout makes it easy to compare the discrepancies between the two graphs.

Viewing and editing notes

Notes are used to track additional information about reports and documents. You can view the notes attached to a report or document in the Notes tab of the Report Data area.

Note the following:

- Users of Integrity Manager can view, add, and edit notes even if they do not have the privileges to view, add, or edit notes in MicroStrategy Web or Developer.
- Notes are not supported on versions of Intelligence Server prior to 9.0. If Integrity Manager connects to an Intelligence Server of version 8.1.2 or earlier, the Notes tab displays the message “Notes are not supported for this connection."
- In a baseline-versus-project or baseline-versus-baseline test, the notes for the baselines can be viewed but not edited.

To make sure you are viewing the most recent version of the notes, click **Refresh**. Integrity Manager contacts Intelligence Server and retrieves the latest version of the notes attached to the report or document.

 © 2016, MicroStrategy Inc.

To add a note, enter the new note and click **Submit**. To edit the notes, click **Edit**, make changes to the listed notes, and click **Submit**.

If a Login dialog box opens, select an authentication method, enter a username and password, and click **OK**.

Grouping differences in graph and PDF reports

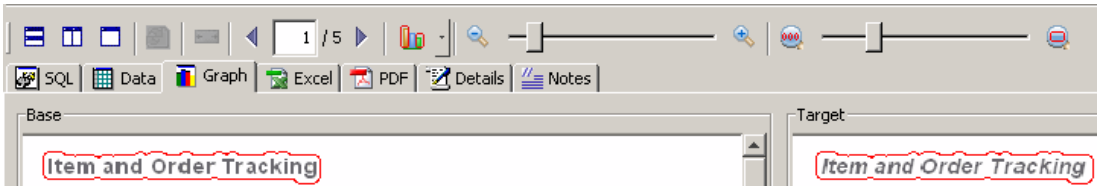
When Integrity Manager compares two graph or PDF reports, it saves the graphs as `.png` or `.pdf` files. It then performs a pixel-by-pixel comparison of the two images. If any pixels are different in the base and target graph, the graph or PDF is considered **Not Matched**.

Adjacent pixel differences are grouped together and treated as a single difference. When you view the graph or PDF reports, Integrity Manager draws a red boundary around the currently selected difference. To navigate through the differences, use the **Next Difference** and **Previous Difference** icons on the Report Data toolbar.

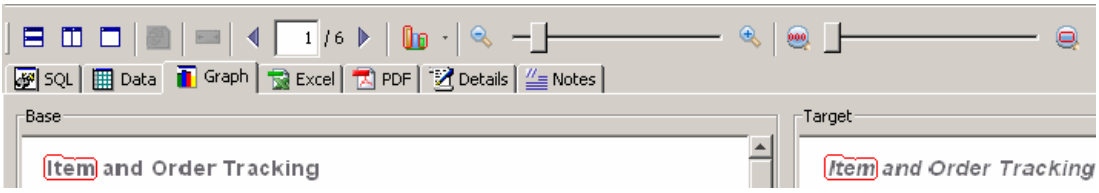
Each difference has a boundary of unchanged pixels that is treated as part of the difference. You can adjust the size of this boundary with the **Granularity** slider on the Report Data toolbar. Increasing the granularity causes multiple differences near each other to be treated as a single difference. This can be useful when you want to treat the changes to the formatting of a title or legend as a single difference, so that you can quickly navigate to any other differences.

In the image below, the title for the graph has been changed between the baseline and the target. In the base graph, the title is in normal font; in the target, it is in italic font.

The white space between the words is the same in both the base and target reports. When the granularity is set to a low level, this unchanged space causes Integrity Manager to treat each word as a separate difference, as seen below:



If the granularity is set to a higher level, the space between the words is no longer sufficient to cause Integrity Manager to treat each word as a separate difference. The differences in the title are all grouped together, as seen below:



© 2016, MicroStrategy Inc.

Accessing the saved results of a test

When you execute a test, Integrity Manager saves the results of that test to a location specified in the Select Execution Settings page of the Integrity Manager wizard. If the option labeled **Store output in a time stamped sub-folder of this directory** is selected, the test results are stored in a subfolder of the specified output folder. Otherwise, the test results are stored directly in the output folder.

A summary of the test results is available in HTML, in the file `ResultsSummary.html`. This file gathers data from the file `ResultsSummary.xml` and formats the data with the stylesheets `style.css` and `ResultsSummary.xsl`.

While the test is executing, a temporary results file, `temp.xml`, is created. This file is updated as each report or document completes execution. In the event of a system crash during test execution, the most recent results are stored in this file.

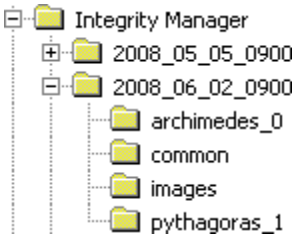
Report execution output

Within the output folder, Integrity Manager creates a folder named `images` to store the images used in the `ResultsSummary` files. For a comparative integrity test, a folder named `common` is created to hold the serialized comparison files.

Integrity Manager also creates a separate folder within the output folder for the report or document results from each project. These folders are named after the Intelligence Server machines on which the projects are kept.

- For the baseline server, `_0` is appended to the machine name to create the name of the folder.
- For the target server, `_1` is appended to the machine name.

For example, the image below is taken from a machine that executes a project-versus-project integrity test at nine AM on the first Monday of each month. The baseline project is on a machine named `ARCHIMEDES`, and the target project is on a machine named `PYTHAGORAS`. The folder for the results from the baseline project is `archimedes_0`, and the folder for the results from the target project is `pythagoras_1`.



In a baseline-versus-project integrity test, the baseline folder is named `baseline_0`. In a baseline-versus-baseline integrity test, the baseline folder is named `baseline_0` and the target folder is named `baseline_1`.

Each results folder contains a number of files containing the results of each report that is tested. These files are named *<ID>_<GUID>.<ext>*, where *<ID>* is the number indicating the order in which the report was executed, *<GUID>* is the report object

 © 2016, MicroStrategy Inc.

GUID, and *<ext>* is an extension based on the type of file. The report results are saved in the following files:

- SQL is saved in plain text format, in the file *<ID>*_*<GUID>*.sql.

  Note: In a comparative integrity test, if you select the Save color-coded SQL differences to an HTML file check box, the SQL is also saved in HTML format, in the file *<ID>*_*<GUID>*.htm. In this file, the SQL that is different from the SQL in the other project's version of the report is highlighted in red.

- Grid data is saved in CSV format, in the file *<ID>*_*<GUID>*.csv, but only if you select the **Save CSV files** check box in the Advanced Options dialog box.
- Graph data is saved in PNG format, in the file *<ID>*_*<GUID>*.png, but only if the report has been saved in Graph or Grid/Graph format.
- Excel data is saved in XLS format, in the file *<ID>*_*<GUID>*.xls, but only if you select the **Save XLS files** check box in the Advanced Options dialog box.
- PDF data is saved in PDF format, in the file *<ID>*_*<GUID>*.pdf.
- Notes are saved in plain text format, in the file *<ID>*_*<GUID>*.notes.txt. This file is created even if the corresponding report does not have notes.

Note the following:

- Only report results for formats requested in the Select Processing Options page during test setup are generated.
- SQL, graph, and PDF data are always saved if they are generated. Grid and Excel data are only saved if you choose to save those results during test creation. Notes are always saved.
- Integrity Manager also creates a file named *<ID>*_*<GUID>*.ser for each report or document. These files contain serialized binary data that Integrity Manager uses when you open a previously saved set of test results, and are not intended for use by end users. These files are stored in the same folder as the test results.

Each results folder also contains a file called baseline.xml that provides a summary of the tested reports. This file is used to provide a baseline summary for baseline-versus-project and baseline-versus-baseline integrity tests.

To open a previously saved set of test results

1. In Integrity Manager, from the **File** menu, select **Open Results**.
2. Browse to the location of the saved test.
3. Select the ResultsSummary.xml file and click **Open**. The saved test results open in Integrity Manager, just as if you had executed the test.

© 2016, MicroStrategy Inc.

List of tags in the integrity test file

When you save an integrity test, it is saved as an XML file, with an extension of .MTC. For instructions on saving or loading an integrity test, see *Saving and loading a test, page 671*.

If needed, you can edit the integrity test file with any XML editor or text editor, such as Notepad. The table below lists all the XML tags in an integrity test file, with an explanation of each tag.

XML Tag	Function
General test information	
`Execution_Mode`	Type of integrity test, as displayed in the ResultsSummary file: • **Project versus Project Integrity Test** • **Baseline versus Project Integrity Test** • **Baseline versus Baseline Integrity Test** • **Single Project Integrity Test** **Note**: This value is for display and localization purposes only.
`Execution_Mode_Value`	Type of integrity test, as executed by Integrity Manager: • **1**: Project versus Project integrity test • **2**: Single Project integrity test • **3**: Baseline versus Project integrity test • **4**: Baseline versus Baseline integrity test
`LocalVersion`	Version of Integrity Manager that created the test.
`isMultiUser`	Whether this integrity test supports multiple logins: • **true**: This integrity test supports multiple logins. • **false**: This integrity test does not support multiple logins.
`Base_Connection_Index`	Which `ConnectionIndex` (0 or 1) indicates the base connection. The other `ConnectionIndex` is the target connection.
Base or Target connection information	
Note: Except in a single project integrity test, this section is repeated for both the base connection and the target connection.	
`ConnectionIndex=`	**0** or **1**, depending on the value of `Base_Connection_Index` and whether the information below is for the base or target connection.
`Server_Name`	Name or IP address of the Intelligence Server.

 © 2016, MicroStrategy Inc.

XML Tag	Function
Port	Port number of the Intelligence Server.
Authentication_Mode	Login authentication mode corresponding to the Login tag below. If isMultiUser is set to **true**, there can be multiple Authentication_Mode and Login tag pairs. • **1**: Standard • **2**: Windows • **16**: LDAP • **32**: Database
Login	Login ID corresponding to the Authentication_Mode tag above. If isMultiUser is set to **true**, there can be multiple Authentication_Mode and Login tag pairs.
Project	Name of the project.
Project_DssID	GUID of the project.
Version	The version of Intelligence Server that hosts the project.
Use_Load_Balancing	Whether to use load balancing across the cluster for this connection, that is, whether to execute the reports/documents across all nodes of the cluster or on a single node: • **true**: Use load balancing. • **false**: Do not use load balancing.
baselineConnection	Whether this connection uses a baseline file: • **true**: This connection uses a baseline. • **false**: This connection uses a live Intelligence Server.
baselineFile	The full path to the baseline file, if baselineConnection is set to true.
Objects to be tested	
Note: This section must be repeated for each object included in the integrity test.	
Type	Type of object to be processed by Integrity Manager: • **3**: Report • **8**: Folder • **18**: Shortcut • **39**: Search object • **55**: Document
GUID	GUID of the object.
Name	Name of the object.

XML Tag	Function
`Path`	Path to the object within the project.
`Rounds`	*(This entry is deprecated.)*
`Reporttype`	Object type. If `Type` is set to 3: • **768**: Grid view report • **769**: Graph view report • **770**: SQL view report • **774**: Grid/Graph view report • **776**: Intelligent Cube • **778**: Transaction • **4096**: Datamart report If `Type` is set to 55: • **14081**: Document
`chaseSearches`	Whether embedded search objects are processed by the integrity test: • **true**: Process embedded search objects. • **false**: Do not process embedded search objects.
`objMatchType`	Whether to match objects by ID or path name: • **0**: Match by ID. • **1**: Match by path name.
`Use_Obj_Match`	Whether object matching is used. This is only available for Project to Project tests: • **true**: Object matching is used. This allows you to select which object from the base project is compared to which object from the target project. • **false**: Object matching is not used.
`Obj_Match_Map`	Only displays if Use_Obj_Match is true. Inside the map is one or more Entry statements which each contain a Key/Value pair for a mapped object where: • Key is the GUID of the object in the base project • Value is the GUID of the object in the target project
Prompt settings	
`textAnswer`	Custom answer for text prompts. To provide multiple custom answers for text prompts, include each answer in a separate `textAnswer` node.

 © 2016, MicroStrategy Inc.

XML Tag	Function
`textAnswerIsNull`	Whether custom answers are provided for text prompts: • **true**: Custom answers are not provided for text prompts. • **false**: Custom answers are provided for text prompts.
`numberAnswer`	Custom answer for numeric or Big Decimal prompts. To provide multiple custom answers for these prompts, include each answer in a separate `numberAnswer` node.
`numberAnswerIsNull`	Whether custom answers are provided for numeric and Big Decimal prompts: • **true**: Custom answers are not provided for numeric and Big Decimal prompts. • **false**: Custom answers are provided for numeric and Big Decimal prompts.
`dateAnswer`	Custom answer for date prompts. To provide multiple custom answers for date prompts, include each answer in a separate `dateAnswer` node.
`dateAnswerIsNull`	Whether custom answers are provided for date prompts: • **true**: Custom answers are not provided for date prompts. • **false**: Custom answers are provided for date prompts.
`numberElementHierPrompt`	Number of elements that Integrity Manager selects to answer element hierarchy prompts.
`numberElementHierPrompt IsNull`	Whether a custom value is provided for the number of elements used to answer element hierarchy prompts: • **true**: A custom value is not provided for element hierarchy prompts. • **false**: A custom value is provided for element hierarchy prompts.
`answerOptionalPrompt`	Whether optional prompts are answered in this integrity test: • **true**: Optional and required prompts are answered. • **false**: Only required prompts are answered.

© 2016, MicroStrategy Inc.

XML Tag	Function
`PromptAnswerSource`	The prompt answer sources to be used by this integrity test, in the order that they are to be used, separated by commas. Negative numbers indicate that this answer source is disabled. **Note**: All four prompt answer sources must be included in this parameter. • **1**: Personal answer • **-1**: Personal answer (disabled) • **2**: Default object answers • **-2**: Default object answers (disabled) • **3**: Integrity Manager user-defined answer • **-3**: Integrity Manager user-defined answer (disabled) • **4**: Integrity Manager internal answer • **-4**: Integrity Manager internal answer (disabled)
`PromptAnswerSource_VAL`	A prompt answer source to be used by this integrity test. If multiple prompt answer sources are specified, each must have its own `PromptAnswerSource_VAL` entry, in the order that they are to be used. Values include: • **Personal answer** • **Default object answers** • **Integrity Manager user-defined answer** • **Integrity Manager internal answer**
`isLinkPopup`	Whether to open reports with unanswered prompts in MicroStrategy Web: • **true**: Open reports with unanswered prompts. • **false**: Do not execute reports with unanswered prompts.
`BaseURL`	If `isLinkPopup` is set to true, the URL for the MicroStrategy Web server for the base project.
`TargetURL`	If `isLinkPopup` is set to true, the URL for the MicroStrategy Web server for the target project.
`Personal_Answer_Option`	Whether to use only default personal prompt answers, or group personal prompt answers by their names: • **USE_DEFAULT**: Use only default personal prompt answers for each prompt. • **GROUP_BY_NAME**: Group personal prompt answers by their names.

 © 2016, MicroStrategy Inc.

XML Tag	Function
Personal_Answer_Option_Desc	If Personal_Answer_Option is set to USE_DEFAULT, this must be set to **Use only default personal prompt answer for each prompt**. If Personal_Answer_Option is set to GROUP_BY_NAME, this must be set to **Group personal prompt answers by their names**.
Execution Settings	
MaxTimeout	Maximum time, in minutes, that a report can run before Integrity Manager cancels it.
numSimultaneous Executions	Maximum number of simultaneous report/document executions during the integrity test.
useCache	Whether to use the cached version of a report, if one is available: • **true**: Use the report cache. • **false**: Do not use the report cache; execute each report against the Intelligence Server.
Output_Directory	Full path to the location where the integrity test results are saved.
isAppendDateToOutputDir	Whether to store the test results in a subdirectory of the Output_Directory, named by date and time of the integrity test execution: • **true**: Store results in a time-stamped subdirectory of the specified directory. • **false**: Store results in the specified directory.
LogLevel	Enable or disable logging: • **1**: Logging is enabled. • **-5**: Logging is disabled.
LogFile	Full path to the log file.
Processing options	
isDataEnabled	Whether to enable data comparison for reports: • **true**: Enabled. • **false**: Disabled.
isSQLEnabled	Whether to enable SQL comparison for reports: • **true**: Enabled. • **false**: Disabled.
isGraphEnabled	Whether to enable graph comparison for reports: • **true**: Enabled. • **false**: Disabled.

XML Tag	Function
isExcelEnabled	Whether to enable Excel comparison for reports: • **true**: Enabled. • **false**: Disabled.
isPdfEnabled	Whether to enable PDF comparison for reports: • **true**: Enabled. • **false**: Disabled.
isRsdExcelEnabled	Whether to enable Excel comparison for documents: • **true**: Enabled. • **false**: Disabled.
isRsdPdfEnabled	Whether to enable PDF comparison for documents: • **true**: Enabled. • **false**: Disabled.
isRsdExecEnabled	Whether to enable execution for documents: • **true**: Enabled. • **false**: Disabled.
reportCycles	Number of performance test cycles to run for each report.
documentCycles	Number of performance test cycles to run for each document.
SQL processing options	
isColorCodeSQL	Whether to save the generated SQL to an HTML file with differences highlighted in red: • **true**: Enabled. • **false**: Disabled.
dynamicSQLStart	Text marking the beginning of any dynamic SQL.
dynamicSQLEnd	Text marking the end of any dynamic SQL.
from	SQL to be replaced by the SQL indicated by the `to` tag.
to	SQL to replace the SQL indicated by the `from` tag.
applyTo	Where to apply the SQL replacement: • **1**: Base only. • **2**: Target only. • **3**: Base and target.
Data processing options	

 © 2016, MicroStrategy Inc.

XML Tag	Function
isCSVEnabled	Whether to save the data for each report as a CSV file: • **true**: Enabled. • **false**: Disabled.
Excel processing options	
Note: For all Excel processing options, if the option is left blank, the setting for that option is imported from the user's MicroStrategy Web export preferences, as per the **Use Default** option in the Integrity Manager Wizard.	
isXLSEnabled	Whether to save data for each report as an XLS file: • **true**: Enabled. • **false**: Disabled.
ExportReportTitle	Whether to include the report title in the Excel chart: • **0**: Do not export the report title. • **-1**: Export the report title.
ExportPageByInfo	Whether to include which report objects are grouped in a page-by selection in the Excel chart: • **0**: Do not export the page-by information. • **-1**: Export the page-by information.
isExportFilterDetails	Whether to include the report filter details in the Excel chart: • **true**: Export the filter details. • **false**: Do not export the filter details.
isRemoveColumn	Whether to remove the extra "Metrics" column from the Excel chart: • **0**: Remove the extra column. • **1**: Do not remove the extra column. • **2**: Use the default setting in the MicroStrategy Web preferences.
ExpandAllPages	Whether to include all report objects in the Excel chart, or only the objects in the default page-by selection: • **0**: Export only the default page-by. • **-1**: Export all objects.
excelVersion	Excel version of the exported file: • **1**: Excel 2000. • **2**: Excel XP/2003. • **4**: Excel 2007 or newer.
isExportMetricAsText	Whether to export metric values as text or as numeric values: • **true**: Export metrics as text. • **false**: Export metrics as numeric values.

© 2016, MicroStrategy Inc.

XML Tag	Function
isExportHeaderAsText	Whether to export data header values as text or as numeric values: • **true**: Export headers as text. • **false**: Export headers as numeric values.
isSeparateSheets	Whether to export each page of the report to a separate sheet in the Excel file: • **true**: Export each page as a separate sheet. • **false**: Export the entire report on a single sheet.
isLiveCharts	Whether to export graphs in the report as live Excel graphs, or as static images: • **true**: Export graphs as live Excel graphs. • **false**: Export graphs as static images in the Results folder.
ExcelEmbedImages	Whether images and graphs in the report can be accessed from the Excel spreadsheet without having to run MicroStrategy Web: • **0**: Images and graphs are not embedded in the spreadsheet, and cannot be accessed without running the report in MicroStrategy Web. • **-1**: Images and graphs are embedded in the spreadsheet.
isOfficeRefresh	Whether MicroStrategy Office can refresh reports after they have been exported to Excel: • **true**: Reports can be refreshed from Office. • **false**: Reports are static and cannot be refreshed from Office.
ExcelReportHeader	Text of the custom header added to the Excel spreadsheet.
ExcelReportHeader Location	The location of the custom header in the Excel export: • **0**: Display the custom header before other report headers. • **1**: Display the custom header after other report headers. • **2**: The custom header replaces any other report headers.
ExcelReportFooter	Text of the custom footer added to the Excel spreadsheet.
PDF processing options	
Note: For all PDF processing options, if the option is left blank or not listed in the MTC file, that option is processed using the default setting in Intelligence Server's PDF generation options.	
Scaling	Whether to adjust the font to fit the report to a certain percentage of the PDF page (ScalePercentage), or to fit a certain number of report pages on the page (ScalePagesWide and ScalePagesTall): • **0**: Use ScalePercentage. • **1**: Use ScalePagesWide and ScalePagesTall.

 © 2016, MicroStrategy Inc.

XML Tag	Function
ScalePercentage	Percentage to scale the font if Scaling is set to 0.
ScalePagesWide	Number of report pages per PDF page width, if Scaling is set to 1.
ScalePagesTall	Number of report pages per PDF page height, if Scaling is set to 1.
GridandGraph	Whether to print the report's grid and graph on the same page: • **0**: Print the grid and graph on separate PDF pages. • **1**: Print the grid and graph on the same PDF page.
Orientation	Page orientation: • **0**: Portrait. • **1**: Landscape.
PrintCoverDetails	Whether to include a cover page: • **0**: Do not print a cover page. • **1**: Print a cover page.
CoverPageDetails Contents	What to include in the cover page, if PrintCoverDetails is set to 1: • **0**: Report filter details. • **1**: Report details.
CoverPageLocation	The location of the cover page, if PrintCoverDetails is set to 1: • **0**: After the report. • **1**: Before the report.
ExpandAllPages	Whether to include all report objects in the PDF, or only objects in the default page-by selection: • **0**: Export only the default page-by. • **1**: Export all objects.
PaperType	Paper size of the PDF: • **0**: Letter (8.5"x11") • **1**: Legal (8.5"x14") • **2**: Executive (7.25"x10.5") • **3**: Folio (8.5"x13") • **4**: A3 (11.69"x16.54") • **5**: A4 (8.27"x11.69") • **6**: A5 (5.83"x8.27")
MarginLeft	Left margin, in inches.

XML Tag	Function
MarginRight	Right margin, in inches.
MarginTop	Top margin, in inches.
MarginBottom	Bottom margin, in inches.
MaxHeaderSize	Maximum header size, in inches.
MaxFooterSize	Maximum footer size, in inches.
GraphFormat	Whether to use bitmaps for graphs: • **10**: Use bitmaps for graphs. • **11**: Do not use bitmaps for graphs.
PrintQuality	Whether to use draft quality bitmaps for graphs, if GraphFormat is set to 11: • **96**: Use draft quality bitmaps. • **288**: Use fine quality bitmaps.
EmbedFonts	Whether to embed fonts in the PDF: • **0**: Do not embed fonts. • **1**: Embed fonts.
HeaderLeft	Left page header.
HeaderCenter	Center page header.
HeaderRight	Right page header.
FooterLeft	Left page footer.
FooterCenter	Center page footer.
FooterRight	Right page footer.
ReportHeader	Report header.
Performance processing options	
inclErrorRptIn Performance	Whether to include in the performance test the execution times from reports/documents that do not complete execution: • **true**: Include the execution times from all reports/documents. • **false**: Include only the execution times from reports/documents that execute successfully.

 © 2016, MicroStrategy Inc.

17

MAINTAINING YOUR MICROSTRATEGY SYSTEM WITH HEALTH CENTER

Introduction

MicroStrategy Health Center can help you prevent, diagnose, and fix problems in your MicroStrategy system. It detects known problems and provides an immediate solution to many of them. Health Center can email a system administrator if it encounters a problem with the system. In cases where Health Center cannot fix a problem immediately, Health Center enables you to bundle relevant system information, such as log files, into a diagnostic package and transmit the package to MicroStrategy Technical Support for review and troubleshooting.

You must have administrative privileges on the Health Center machine to be able to run the Health Center Console or the Configuration Wizard.

This chapter provides the following information on using Health Center:

- *Accessing the Health Center console, page 700*
- *Configuring and managing a Health Center system, page 700*
- *Diagnosing and fixing problems, page 708*
- *Collecting and sending information for MicroStrategy Technical Support, page 711*
- *Keeping Health Center up to date, page 712*

© 2016, MicroStrategy Inc.

- *List of system checks, page 713*
- *Using a response file to configure a Health Agent or Master Health Agent, page 722*

Accessing the Health Center console

You access most Health Center functionality through the Health Center console. To start the Health Center Console:

- In Windows, from the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Health Center Console**.

  If Health Center Console does not start, check to see if this machine has been renamed since installing Health Center. When Health Center Console is launched, it checks for a folder named `machinename-GUI` in the folder where Health Center is installed. The `machinename` is the fully qualified machine name. By default, the installation folder is `C:\Program Files (x86)\Common Files\MicroStrategy\Health Center`. If a folder whose name matches this machine's name is not found, Health Center does not start. If your MicroStrategy system was upgraded, the folder name may have changed. To start Health Center, rename the folder so that it uses the current name of the machine.

- In UNIX/Linux, in the Health Center installation directory, in the `/bin` folder, run the program `mstrsuppappgui`.

Many of the system checks in the Health Center console can be accessed and configured in MicroStrategy Operations Manager—a web-based console. You can also configure alerts in Operations Manager to send an email if certain thresholds are met or exceeded. For background information on Operations Manager, instructions for accessing system check results, and how to configure alerts and monitors, see the *MicroStrategy Operations Manager Guide*.

Configuring and managing a Health Center system

Health Center uses a distributed system architecture. It consists of a central machine (the Master Health Agent) that connects to multiple additional machines (Health Agents).

A Health Center system represents a network of machines that are all administered by the same person or set of people. You can have multiple Health Center systems in a

© 2016, MicroStrategy Inc.

MicroStrategy system. For example, if the machines in the development environment all have one set of administrators, and the machines in the production environment are administered by a different group of people, the development environment and production environment should be configured as separate Health Center systems, each with their own Master Health Agent.

Best practices for configuring a Health Center system

MicroStrategy recommends the following best practices for configuring your Health Center systems:

- Create a separate Health Center system for each group of machines administered by the same people. For example, if your network has one set of administrators for the development machines and another set of administrators for the production machines, the development machines should be in a separate Health Center system from the production machines. This allows you to designate the people who already have administrative access to those machines as Health Center administrators.
- When choosing a machine to be a Master Health Agent, follow the best practices provided in *Best practices for selecting a Master Health Agent machine, page 702*.
- Configure all machines that are running MicroStrategy server products, such as Intelligence Server, MicroStrategy Web, MicroStrategy Mobile Server, Enterprise Manager, or Narrowcast Server, as Health Agents, and make sure they are included in a Health Center system.
- You can also configure all machines running client products, such as Developer or MicroStrategy Web, as Health Agents, and add them to a Health Center system. However, this can result in a very large Health Center system. Depending on the number of client machines in your network, it may be more effective to configure those machines as Health Agents and add them, as needed, to the Health Center system for troubleshooting.
- When you install MicroStrategy products on a machine, specify a unique Health Center access code for that machine, and make sure that only Health Center administrators know the access code. This helps to keep your system secure.
- Each Health Agent/Master Health Agent machine must have a dedicated network port available for use by Health Center. If the specified port is in use by another application, the Health Agent/Master Health Agent service will not start.
- If you are configuring Health Agents/Master Health Agents using a network account, make sure that account has read and write access to the Health Center installation location on the machines.
- Configure UNIX or Linux Health Agents/Master Health Agents as daemons, so that the Health Agent/Master Health Agent process is constantly running in the background. This requires you to configure the Health Agent/Master Health Agent using an account that has root access to that machine. If you do not have root access to the machine, you can still configure the Health Agent/Master Health Agent as an application. In this case, be careful not to stop the Health Agent/Master Health Agent process, so that the machine can remain part of the Health Center system at all times.

© 2016, MicroStrategy Inc.

- Participate in the MicroStrategy Customer Experience Improvement program so that MicroStrategy can better serve your needs as a customer.

The Master Health Agent

The Master Health Agent machine serves as the hub of a Health Center system. It keeps track of the scheduled system checks for itself and the connected Health Agents, and it triggers those system checks at the appropriate times. It also stores the network topology for the system and transmits diagnostic packages to MicroStrategy Technical Support.

Best practices for selecting a Master Health Agent machine

MicroStrategy recommends the following best practices for choosing a Master Health Agent machine:

- The machine should be always available on the network so that users can run diagnostic tests at any time.
- The machine should have HTTPS access so that it can download Health Center updates.
- The machine should have FTP access so that it can send diagnostics packages to MicroStrategy Technical Support. For information about diagnostics packages, see *Collecting and sending information for MicroStrategy Technical Support, page 711*.
- The machine should not have a heavy additional workload. For example, a production environment Intelligence Server machine may not make a good Master Health Agent.

Configuring a machine as a Master Health Agent

Use the Configuration Wizard to configure a machine as a Master Health Agent.

You can automate the configuration of the Master Health Agent by using a Configuration Wizard response file. For detailed instructions, see *Using a response file to configure a Health Agent or Master Health Agent, page 722*.

Prerequisites

- The Master Health Agent machine must have a dedicated network port available for use by Health Center. If the specified port is in use by another application, the Master Health Agent service does not start.
- In a UNIX or Linux system, the Master Health Agent and daemon must be configured by a user with root access.

 © 2016, MicroStrategy Inc.

To configure a machine as a Master Health Agent

1. Open the Configuration Wizard:
   - In Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Configuration Wizard**. The Configuration Wizard opens with the Welcome page displayed.
   - In UNIX/Linux: From a UNIX/Linux console window, navigate to the MicroStrategy target directory (the path indicated during installation as the Directory Name). Then navigate to the bin directory and type `./mstrcfgwiz` at the prompt. The Configuration Wizard opens with the Welcome page displayed.
2. Select **Health Center Configuration** and click **Next**. The Health Center Configuration page opens.
3. Select **Set up a Master Health Agent** and click **Next**. The Connection Parameters page opens.
4. Step through the wizard, providing any information required. For information about a page of the wizard, click **Help** on that page.
5. Click **Finish**. The machine is configured as a Master Health Agent, and the Health Agent service is started.

Health Agents

Any machine running MicroStrategy software can be configured as a Health Agent. Health Agents perform system checks on themselves when scheduled to do so by the Master Health Agent, and they report the results of those checks to the Master Health Agent.

Because the Health Agent process requires relatively few system resources, you can configure all machines in your network as Health Agents with minimal effect on performance.

Configuring a machine as a Health Agent

Use the Configuration Wizard to configure a machine as a Health Agent.

You can automate the configuration of the Health Agent by using a Configuration Wizard response file. For detailed instructions, see *Using a response file to configure a Health Agent or Master Health Agent, page 722*.

Prerequisites

- The Health Agent machine must have a dedicated network port available for use by Health Center. If the specified port is in use by another application, the Health Agent service does not start.

- In a UNIX or Linux system, the Master Health Agent and daemon must be configured by a user with root access.

To configure a machine as a Health Agent

1. Open the Configuration Wizard:
   - In Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Configuration Wizard**. The Configuration Wizard opens with the Welcome page displayed.
   - In UNIX/Linux: From a UNIX/Linux console window, navigate to the MicroStrategy target directory (the path indicated during installation as the Directory Name). Then navigate to the bin directory and type `./mstrcfgwiz` at the prompt. The Configuration Wizard opens with the Welcome page displayed.
2. Select **Health Center Configuration** and click **Next**. The Health Center Configuration page opens.
3. Select **Set up a Health Agent** and click **Next**. The Connection Parameters page opens.
4. Step through the wizard, providing any information required. For information about a page of the wizard, click **Help** on that page.
5. Click **Finish**. The machine is configured as a Health Agent and the Health Agent service is started.

Adding Health Agents to the system

When you add a Health Agent to the Health Center system, the Master Health Agent can perform tasks such as scheduling system checks or viewing log files for that machine.

You can use the Configuration Wizard to add a local Health Agent to the network. You can add additional Health Agents to the system individually, or you can use Health Center to discover machines that are running MicroStrategy components elsewhere in your MicroStrategy network. All these procedures are described below. For instructions on how to configure a machine as a Health Agent, see *Configuring a machine as a Health Agent, page 703*.

If a Health Agent machine is behind a firewall, you must use Configuration Wizard to add it to the Health Center system. For instructions, see *Adding a Health Agent from behind a firewall, page 706*.

To add the local machine to the Health Center system

1. On that machine, open the Configuration Wizard:

 © 2016, MicroStrategy Inc.

- In Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Configuration Wizard**. The Configuration Wizard opens with the Welcome page displayed.
- In UNIX/Linux: From a UNIX/Linux console window, navigate to the MicroStrategy target directory (the path indicated during installation as the Directory Name). Then navigate to the bin directory and type `./mstrcfgwiz` at the prompt. The Configuration Wizard opens with the Welcome page displayed.

2. Select **Health Center Configuration** and click **Next**. The Health Center Configuration page opens.
3. Select **Connect local Health Agent to a Health Center system** and click **Next**. The Master Health Agent Connectivity Parameters page opens.
4. In the **Master Health Agent**, **Port Number**, and **Access Code** fields, type the machine name, port number, and Health Center access code for the Master Health Agent. If the Master Health Agent does not have an access code, leave this field blank.
5. If the local machine is behind a firewall, you may need to specify additional information. For details, click **Help**, or see *Adding a Health Agent from behind a firewall, page 706*.
6. Click **Next**. The Summary page opens. Review the information on this page and click **Finish**. The machine is added to the Health Center network as a Health Agent.

To add a remote Health Agent to the Health Center network

1. In the Health Center Console, in the **System** panel, right-click the Master Health Agent and select **Add**. The Add Machine dialog box opens.
2. In the **Machine** and **Port Number** fields, type the name and port number of the Health Agent machine. If the Health Agent was configured to require an access code, type that access code in the **Access Code** field.
3. Click **Add**. The dialog box closes and the machine is added to the network. It is now visible in the network topology view in the main pane.

To discover Health Agents in your MicroStrategy network

1. In the Health Center Console, in the **System** panel, right-click a machine and select **Discover Machines**. The Machine Discovery Wizard opens.
2. Step through the wizard, providing any required information. For instructions on each page of the wizard, click **Help**.
3. When you reach the Summary page, select the machines that you want to add to the Health Center system and click **Finish**. The wizard closes and an Add Machine dialog box opens for each machine selected.

© 2016, MicroStrategy Inc.

If a machine has been configured as a Health Agent and is using the default port and does not require an access code, an Add Machine dialog box does not open for that machine. Instead, that machine is automatically added to the Health Center system.

4 For each machine, in the Add Machine dialog box, type the port number and access code (if required) and click **OK**. That machine is added to the Health Center system.

Adding a Health Agent from behind a firewall

If a Health Agent machine is behind a firewall from the Master Health Agent, you may encounter difficulty adding the Health Agent to the Health Center system. In this case, you must add the Health Agent to the system from the Health Agent machine, using the Configuration Wizard.

One common scenario is a single firewall around a group of several Health Agent machines. In this case, depending on your network topology, you may want to connect one of the Health Agents directly to the Master Health Agent and route the other Health Agents through that Health Agent. Instructions for this procedure are below.

If the Master Health Agent machine is down when you attempt to add the local machine to the Health Center system through a firewall, when it is running again you must configure the Master Health Agent to await an incoming connection. Instructions for this procedure are below.

To add a local Health Agent that is behind a firewall to the Health Center system

1 On that machine, open the Configuration Wizard:

- In Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Configuration Wizard**. The Configuration Wizard opens with the Welcome page displayed.
- In UNIX/Linux: From a UNIX/Linux console window, navigate to the MicroStrategy target directory (the path indicated during installation as the Directory Name). Then navigate to the bin directory and type `./mstrcfgwiz` at the prompt. The Configuration Wizard opens with the Welcome page displayed.

2 Select **Health Center Configuration** and click **Next**. The Health Center Configuration page opens.

3 Select **Connect local Health Agent to a Health Center system** and click **Next**. The Health Agent Connectivity Parameters page opens.

4 In the **Master Health Agent**, **Port Number**, and **Access Code** fields, type the machine name, port number, and Health Center access code for the Master Health Agent. If the Master Health Agent does not have an access code, leave this field blank.

5 Click **Advanced**. The Advanced Options become visible.

 © 2016, MicroStrategy Inc.

6 Select **Establish outbound connection**.

7 Click **Next**. The Summary page opens. Review the information on this page and click **Finish**. The machine is added to the Health Center network as a Health Agent.

To add a local Health Agent to the Health Center system through another Health Agent

1 On that machine, open the Configuration Wizard:

- In Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Configuration Wizard**. The Configuration Wizard opens with the Welcome page displayed.
- In UNIX/Linux: From a UNIX/Linux console window, navigate to the MicroStrategy target directory (the path indicated during installation as the Directory Name). Then navigate to the bin directory and type `./mstrcfgwiz` at the prompt. The Configuration Wizard opens with the Welcome page displayed.

2 Select **Health Center Configuration** and click **Next**. The Health Center Configuration page opens.

3 Select **Connect local Health Agent to a Health Center system** and click **Next**. The Master Health Agent Connectivity Parameters page opens.

4 In the **Master Health Agent**, **Port Number**, and **Access Code** fields, type the machine name, port number, and Health Center access code for the Master Health Agent. If the Master Health Agent does not have an access code, leave this field blank.

5 Click **Advanced**. The Advanced Options become visible.

6 If there is a firewall between this machine and the Health Agent you are connecting through, select **Establish outbound connection**.

7 In the **Health Agent** and **Port Number** fields, type the machine name and port number for the Health Agent machine you are connecting through.

8 Click **Next**. The Summary page opens. Review the information on this page and click **Finish**. The machine is added to the Health Center network as a Health Agent, connected through the specified Health Agent.

To configure the Master Health Agent to connect to a Health Agent through a firewall

1 In the Health Center Console, in the System panel, right-click the Master Health Agent and select **Await incoming connection**. The Await Incoming Connection dialog box opens.

2 In the **Health Agent name** field, type the name of the Health Agent machine that is trying to connect to this Master Health Agent. If the Health Agent was configured to require an access code, type that access code in the **Access Code** field.

3 Click **OK**. The Health Agent can now connect to this Master Health Agent.

Diagnosing and fixing problems

One of the main purposes of Health Center is to diagnose any problems with your MicroStrategy system. Health Center does this by performing system checks on the Master Health Agent and on all Health Agents in the Health Center system to identify any problems. In many cases, Health Center also provides you with the ability to fix the problems immediately.

You can execute these system checks manually from the Health Center Console, or you can schedule them to automatically run daily or weekly at a specified time. The Master Health Agent handles the scheduling, and at the appropriate times it triggers the scheduled system checks on all Health Agents in the system.

For a list of all system checks available through Health Center, see *List of system checks, page 713*.

To manually execute a system check

1 In the Health Center Console, in the **System Checks** panel, expand the machine you want to run a system check on, expand the category containing that system check, and select that system check. For detailed information about each system check, see *List of system checks, page 713*.

2 If the system check is displayed in blue, it may require additional configuration information. In the **Configuration** tab, type the required information.

3 In the navigation pane on the left, right-click the system check and select **Refresh**. The system check is performed immediately.

To schedule system checks for a machine

1 In the Health Center console, in the **System Checks** panel, select the machine to schedule system checks for.

2 In the main pane, select the **Scheduling** tab.

Not all system checks can be scheduled to run at all times. Some can be scheduled to run daily, and some can be scheduled to run weekly. The

 © 2016, MicroStrategy Inc.

documentation for each individual system check specifies when it can be scheduled.

3 To schedule daily system checks:

   a Select the **Daily System Checks** check box.

   b Specify a time for the system checks to be executed. The Master Health Agent initiates the system checks at this time every day according to its system clock.

   c Select the check boxes next to the system checks that you want to run daily. For details on each system check, see *List of system checks, page 713*.

4 To schedule weekly system checks:

   a Select the **Weekly System Checks** check box.

   b Specify a day of the week for the system checks to be executed.

   c Specify a time for the system checks to be executed. The Master Health Agent initiates the system checks at this time on the specified day according to its system clock.

   d Select the check boxes next to the system checks that you want to run weekly. For details on each system check, see *List of system checks, page 713*.

5 To email a system administrator if a scheduled system check cannot be completed, select the **Notify the following email accounts in case the System Check is unable to run** check box and type the administrator's email address in the field. To include more than one email address, separate them with commas.

6 To email a system administrator if a scheduled system check reports a problem, select the **Notify the following email accounts in case the System Check detects a problem** check box and type the administrator's email address in the field. To include more than one email address, separate them with commas.

7 In the lower right of the main pane, click **Save**. The schedule for this machine is saved.

Fixing problems found by Health Center

When a system check encounters a problem with a Health Agent, that system check is displayed in red in the navigation pane in the **System Checks** panel. You can view detailed information about the problem by selecting it in the navigation pane. The **Current Status** tab in the main pane of the Health Center console displays information about the system check, and how and why it failed.

In many cases, Health Center can help you resolve the problem. Click the hyperlink provided in the **Current Status** tab to correct the issue. For example, if the Path Environment Variable system check fails, the **Current Status** tab provides you with a hyperlink that you can click to add the MicroStrategy Common Files directory to the system path.

You can view a list of all actions you have taken through the Health Center Console in the **Action History** panel.

© 2016, MicroStrategy Inc.

Other problems may not be able to be fixed from within Health Center. In these cases the **Current Status** tab indicates what the problem is and possible actions you can take to resolve the problem.

Analyzing log files

In the Health Center Console, you can review the MicroStrategy log files to get more information about the possible causes of any errors in your MicroStrategy system. For example, if Intelligence Server returns an Out Of Memory error, you can look in the DSSErrors.log file to see what tasks have used more memory than expected.

All MicroStrategy system components record information about low-level system activity in log files. This information can include system errors, memory usage, starting and stopping services, and so on.

The log files for all the machines in your Health Center network are listed on the **Log Analysis** panel. To view the contents of a log file in the main pane of the Health Center console, double-click that file.

Most log files are displayed as plain text. However, certain log files, such as `DSSErrors.log` or `JVMMessages.log`, are displayed in a format that can be analyzed in greater detail.

To view the contents of a log file

1. In the Health Center Console, in the **Log Analysis** panel, expand the Health Agent containing the log file that you want to view.
2. Expand the MicroStrategy component containing the log file that you want to view.
3. Double-click the log file that you want to view. The log file opens in the main pane of the Health Center Console.
4. If the log file is a MicroStrategy log file such as `DSSErrors.log` or `JVMMessages.log`, you can sort and filter it, as follows:
   - To sort the log entries by a column, click that column header. To reverse the sort, click the column header again.
   - To show or hide columns, right-click a column header and select or clear the columns to display or hide.
   - To rearrange the columns, click and drag the column headers.
   - To view the contents of a row, select a cell in that row. The contents of that row are displayed in the **Entry Details** tab at the bottom of the page. The selected cell is displayed in the **Selected Cell** field in the **Entry Details** tab.
   - To filter the log file entries, or to highlight specific entries, use the Filter pane above the grid. For detailed instructions on how to filter or highlight entries, see the Health Center Help (from within the Health Center console, press **F1**).

 © 2016, MicroStrategy Inc.

Collecting and sending information for MicroStrategy Technical Support

If a problem in the system cannot be resolved by the actions suggested by Health Center, the next step is often to contact MicroStrategy Technical Support. Health Center can collect all the relevant information into a single diagnostics package, which can then be delivered to MicroStrategy Technical Support.

A diagnostics package contains system information, including log files, collected from one or more Health Agent machines. When you create a diagnostics package, you can choose what machines to include log files from, and what log files to include in the package. For more information about log files, see *Analyzing log files, page 710*.

Once you have created a diagnostics package, you can save it as a ZIP file, or send it directly to MicroStrategy Technical Support over FTP. You can use either standard FTP mode, or an encrypted FTP mode (FTPS) that makes use of SSL.

Health Center does not support transmitting packages using secure FTP (SFTP), that is, FTP over SSH.

To create and send a diagnostics package

1. In the Health Center Console, select the **System** panel.
2. In the main pane, right-click the machine that you want to create a diagnostics package for and select **Collect Diagnostics**. The Diagnostics Information Collection Wizard opens at the Welcome page.
3. Step through the Diagnostics Information Collection Wizard and type the information requested. For more information about each page of the wizard, click **Help** on that page.
4. If you do not wish to immediately save the diagnostics package to a ZIP file or transmit it to MicroStrategy Technical Support, click **Finish**. The wizard closes, and your diagnostics package is saved.
5. To save the diagnostics package to a ZIP file:
   a. On the Summary and Transmission page, click **Send**. The Diagnostics Package Summary dialog box opens.
   b. From the **Destination** drop-down list, select **Master Health Agent Repository**.
   c. Click **Send**. The package is saved in the Health Center repository, in the `FileDestination` subfolder.
6. To transmit the diagnostics package to MicroStrategy Technical Support via FTP or FTPS:

© 2016, MicroStrategy Inc.

a On the Summary and Transmission page, click **Send**. The Diagnostics Package Summary dialog box opens.

b From the **Destination** drop-down list, select either **Tech Support FTP Site** for normal FTP, or **Tech Support FTPS Site** for FTP using SSL.

c To modify the destination, click **Modify**. The FTP / FTPS Destination dialog box opens. Make any changes necessary and click **OK**. For more information about the options available, click **Help**.

d Click **Send**. The Diagnostics Package Summary dialog box closes, and the package is transmitted to MicroStrategy Technical Support.

Keeping Health Center up to date

MicroStrategy is constantly extending the functionality of Health Center by developing new system checks and information collectors and refining existing ones. Using the Update feature, Health Center can connect to MicroStrategy's servers to download the latest system checks and other Health Center plug-ins.

Health Center checks for updates every week, beginning 24 hours after you configure the Master Health Agent. If updates are available, the Health Center Console notifies you at the bottom of the main pane. You can then review the available updates and download them to the Master Health Agent.

If your MicroStrategy maintenance contract has expired, you can see that new updates exist, but you will not be able to download those updates until you renew your maintenance contract. For information about renewing your maintenance contract, contact your MicroStrategy account representative.

The Master Health Agent applies the updates as soon as the download is complete. To do this, the Master Health Agent service (in Windows) or daemon (in UNIX/Linux) stops and restarts.

Once the Master Health Agent has been updated, it pushes the updates out to the Health Agents in the system. This causes the Health Agent service or daemon to stop and restart to apply the updates.

To download and apply the latest Health Center updates

1 On the Master Health Agent machine, in the Health Center Console, from the **File** menu, point to **Health Center Update**, and select **Update from MicroStrategy Web**. The Health Center Updates dialog box opens.

2 Review the list of new, updated, and deprecated plug-ins.

 © 2016, MicroStrategy Inc.

3 To download new plug-ins, ensure that the check box next to each new plug-in is selected.

- Updates for existing plug-ins will be automatically downloaded and applied.
- Deprecated plug-ins will be automatically removed.

4 To download the updates, click **Download**. The updates are downloaded and applied to the Master Health Agent and are then pushed out to all Health Agents in the Health Center system.

Manually updating your Health Center system

In some environments, the Master Health Agent machine may not be able to access the MicroStrategy website to download the Health Center updates. For instance, the entire Health Center system, including the Master Health Agent, may be behind a firewall. In this case you may need to manually update your Health Center system.

You can download the latest Health Center update by contacting MicroStrategy Technical Support. You can then apply the update from within the Health Center console.

To manually download and apply the latest Health Center updates

1 Contact MicroStrategy Technical Support to download the latest Health Center update. The update consists of two files:

- `HealthCenterUpdate.zip`
- `HealthCenterUpdateVersionInfo.xml`

2 Copy these files to a location on your network that the Master Health Agent can access.

3 On the Master Health Agent machine, in the Health Center Console, from the **File** menu, point to **Health Center Update**, and select **Update from File**. An Open dialog box opens.

4 Browse to the folder containing the Health Center update. Select this folder and click **Open**. The Health Center update is applied to the Master Health Agent and is then pushed out to all Health Agents in the Health Center system.

List of system checks

A system check is a Health Center scan of a component of a MicroStrategy product. System checks can report errors or potential errors and, in many cases, can provide ways

to fix these problems.

Health Center supports the following system checks:

- *System checks for Configuration, page 715*
  - *C++ Runtime Libraries, page 715*
  - *Logging Settings, page 715*
- *System checks for Developer, page 715*
  - *Component Registration, page 715*
  - *LogPath Registry Key Validation, page 715*
  - *Metadata Connection Validation, page 716*
- *System checks for Enterprise Manager, page 716*
  - *ETL Service Status and Data Load Failures, page 716*
- *System checks for Intelligence Server, page 716*
  - *Component Registration, page 716*
  - *Directories Available Disk Space, page 717*
  - *Directories Permissions, page 717*
  - *Logging Settings, page 717*
  - *LogPath Registry Key Validation, page 717*
  - *Memory Depletion, page 718*
  - *Memory Governing Settings, page 718*
  - *Metadata Connection Validation, page 719*
  - *Network Port Status and Usage Detector, page 719*
  - *Path Environment Variable Validation, page 719*
  - *Warehouse Connections Validation, page 719*
- *System checks for Narrowcast Server, page 720*
  - *Component Registration, page 720*
  - *DCOM Settings, page 720*
  - *Logging Settings, page 720*
  - *Network Ports, page 720*
  - *Object Repository Connection, page 721*
  - *Path Environment Variable Validation, page 721*
- *System checks for MicroStrategy Web products, page 721*

 © 2016, MicroStrategy Inc.

- *Available Memory for JVM setting, page 721*
- *Directories Permissions, page 721*
- *JVM Maximum Heap Size, page 722*
- *Temporary Folder Available Disk Space, page 722*

System checks for Configuration

C++ Runtime Libraries

This system check verifies that the C++ runtime libraries have been updated with the security update to Visual Studio 2010 Service Pack 1. This is required by the Health Center Update system.

This system check is available for Windows machines.

If the most recent update has not been installed, Health Center prompts you to install it.

Logging Settings

This system check allows you to change what information is logged from various MicroStrategy products, such as Developer, Command Manager, Health Center, Object Manager, and Narrowcast Server. You can also specify whether to log information to a MicroStrategy log file or to the MicroStrategy console.

To use the default logging settings for MicroStrategy products on this machine, under that server instance click **Click to use default machine configuration**.

System checks for Developer

Component Registration

This system check verifies that all .EXE, .DLL, and .OCX files required by Developer are present and properly registered on this machine.

This system check is available for Windows machines.

If any files are missing, Health Center provides a list of those files. Contact MicroStrategy Technical Support for information about how to reinstall the missing files.

If any files are not registered correctly, click the link provided to register them.

LogPath Registry Key Validation

This system check confirms that the LogPath registry key exists and is pointing to a valid location for Developer log files.

If the registry key points to an invalid location, you can modify the log path by clicking **Modify MicroStrategy Developer Log Path**. Type the log path in the field and click **OK**.

Metadata Connection Validation

This system check validates the metadata connection information used by all direct (two-tier) project sources on this machine. It performs the following actions:

- Connects to the metadata database
- Verifies that a metadata is present in the database
- Verifies that the metadata uses the correct database prefixes

If Health Center detects any problems with the metadata connection, Health Center instructs you to launch the MicroStrategy Configuration Wizard.

System checks for Enterprise Manager

ETL Service Status and Data Load Failures

This system check analyzes the Enterprise Manager data load process. It performs the following actions:

- Scans the data load log file and reports whether that file contains any errors
- Checks to see if the data load process has been running for longer than the maximum time specified in the configuration
- Verifies that the data load service is running

In the **Configuration** tab, specify the number of hours a data load can proceed before this system check reports an error. The default is 5 hours.

If the data load service is on this machine but not running, click the link provided to restart it.

System checks for Intelligence Server

Component Registration

This system check verifies that all .EXE, .DLL, and .OCX files required by Intelligence Server are present and properly registered on this machine.

This system check is available for Windows machines.

If any files are missing, Health Center provides a list of those files. Contact MicroStrategy Technical Support for information about how to reinstall the missing files.

If any files are not registered correctly, click the link provided to register them.

 © 2016, MicroStrategy Inc.

Directories Available Disk Space

This system check verifies that enough disk space is available for Intelligence Server to function properly. It checks the available disk space on the machine that hosts the Home, Log, Temp, Inbox, Cube, and Cache directories, as specified in the Intelligence Server configuration.

If any of the directories have less space available than the minimum values specified on the **Configuration** tab, either in MB or as a percentage of total disk space, this system check fails. To resolve this problem, either some files should be deleted from the machine hosting the directory, or the system administrator should select a different storage location with more available space.

On the **Configuration** tab, specify a user name and password and authentication mode to log into Intelligence Server, and a port number to use to access Intelligence Server. You can also specify the minimum free disk space that must be available for each of the directories, either in MB or as a percentage of total disk space or both. The system check fails if either condition is not met. A value of -1 indicates no limit. The default values for all the directories are 1024 MB and 10 percent of total disk space.

Directories Permissions

This system check verifies that the network account used to run Intelligence Server can read from and write to the Home, Log, Inbox, Cube, and Cache storage directories.

If that account does not have read and write permissions for the directories, this system check fails. To resolve this problem, the network administrator should grant read and write permissions for those directories to the account being used, or the system administrator should select a different location for Intelligence Server to use so that the account does have read and write permissions.

On the **Configuration** tab, specify a user name and password and authentication mode to log into Intelligence Server, and a port number to use to access Intelligence Server.

Logging Settings

This system check allows you to change what information is logged. You can specify whether to log information to the Windows event log or to a MicroStrategy log file.

To change the default logging settings for all server instances on this machine, click **Change default machine configuration**.

To use the default logging settings for a server instance, under that server instance click **Click to use default machine configuration**.

To change the logging settings for a server instance, under that server instance click **Change server instance configuration**.

LogPath Registry Key Validation

This system check confirms that the LogPath registry key exists and is pointing to a valid location for Intelligence Server log files.

If the registry key points to an invalid location, click one of the links provided:

© 2016, MicroStrategy Inc.

- To modify the log path, click **Modify MicroStrategy Intelligence Server Log Path**. Type the log path in the field and click **OK**.
- To modify the log path and restart Intelligence Server, click **Modify MicroStrategy Intelligence Server Log Path and restart Intelligence Server**. Type the log path in the field and click **OK**.

Memory Depletion

This system check analyzes the Intelligence Server log files for memory depletion errors.

On the **Configuration** tab, specify a user name and password and authentication mode to log into Intelligence Server, and a port number to use to access Intelligence Server. You must also specify how long, in milliseconds, to keep trying to read the `DSSErrors.log` file if it is being written to by Intelligence Server. The default length is 60000 milliseconds.

If memory depletion errors are detected, you can create a diagnostics package from this window.

Memory Governing Settings

This system check analyzes Intelligence Server for potential memory depletion problems. It performs the following actions:

- Checks to see if Intelligence Server's Memory Contract Manager is enabled and properly configured
- Checks to see if the operating system has allocated enough RAM swap space for Intelligence Server to perform adequately
- On Windows systems, checks to see if 4GT mode is enabled, which is relevant for 32-bit Windows

On the **Configuration** tab, specify a user name and password and authentication mode to log into Intelligence Server, and a port number to use to access Intelligence Server. You must also specify the minimum RAM swap multiplier. The default value is 2.

Note the following:

- The user must have read and write permissions for the Server Definition.
- Standard, Database, and LDAP authentication modes are supported.

The following actions may be available depending on what problems are discovered:

- If Memory Contract Manager is not enabled, click the link provided to enable it. For information about Memory Contract Manager, including configuration instructions, see *Governing Intelligence Server memory use with Memory Contract Manager, page 316*.
- If the memory swap space is less than that required by the specified RAM swap multiplier, type the new amount of swap space in the field provided.

 © 2016, MicroStrategy Inc.

- On 32-bit Windows machines, if 4GT mode is not enabled, click the link provided to enable it.

Metadata Connection Validation

This system check validates the metadata connection information used by Intelligence Server. It performs the following actions:

- Connects to the metadata database
- Verifies that a metadata is present in the database
- Verifies that the metadata uses the correct database prefixes

If Health Center detects any problems with the metadata connection, Health Center instructs you to launch the MicroStrategy Configuration Wizard.

Network Port Status and Usage Detector

This system check analyzes the Intelligence Server machine to determine if the network ports used by MicroStrategy services are open and available. It checks the following ports:

- The port used by Intelligence Server, which can be set in the Project Source Manager. The default port is 34952.
- The port used by Licensing, which is port 8888.
- The port used by Listener, which is port 30172.

If any of these ports are determined to be closed, Health Center notifies you and recommends that you open those ports.

This system check is primarily intended for use during server configuration, and it cannot be scheduled.

Path Environment Variable Validation

This system check validates that the path environment variable includes the location of the MicroStrategy Common Files, so that Intelligence Server can access those files.

This system check is available for Windows machines.

If the path environment variable does not include the location of the common files, you can add the common files to the path by clicking the link provided.

Warehouse Connections Validation

This system check verifies that each DSN connection to a data warehouse, as used in a DBConnection in the Intelligence Server metadata, exists on the Intelligence Server machine.

© 2016, MicroStrategy Inc.

If a warehouse connection is not present, Health Center instructs you to launch the MicroStrategy Connectivity Wizard.

System checks for Narrowcast Server

Component Registration

This system check verifies that all .EXE, .DLL, and .OCX files required by Narrowcast Server are properly registered on this machine.

This system check is available for Windows machines.

If any files are missing, Health Center provides a list of those files. For information about how to reinstall the missing files, contact MicroStrategy Technical Support.

If any files are not registered correctly, click the link provided to register them.

DCOM Settings

This system check verifies that all Windows users that create DCOM objects on behalf of Narrowcast Server have the appropriate DCOM permissions.

If any Windows users are missing DCOM permissions, click the link provided to grant the required privileges to those users.

Logging Settings

This system check allows you to change what information is logged.

To change the logging level for Narrowcast Server, click **Change Narrowcast log level**.

To change the logging level for Narrowcast Server and restart the Narrowcast Server services, click **Change Narrowcast log level and restart services**.

Network Ports

This system check analyzes the Intelligence Server machine to determine if the network ports used by Narrowcast Server are open and available. It checks the following ports:

- The Receiver port, port 20009, if the machine is configured as a Narrowcast Logging Client.
- The Relay port, port 20020, if the machine is configured as a Narrowcast Logging Server.
- The Consumer port, port 20030, if the machine is configured as a Narrowcast Logging Server.
- The Buffer port, port 20040, if the machine is configured as a Narrowcast Logging Server.

 © 2016, MicroStrategy Inc.

If any of these ports are determined to be closed, Health Center notifies you and recommends that you open those ports.

This system check is primarily intended for use during server configuration, and it cannot be scheduled.

Object Repository Connection

This system check validates the object repository database connection information used by Narrowcast Server. It performs the following actions:

- Connects to the object repository database
- Verifies that an object repository is present in the database
- Verifies that the object repository uses the correct database prefixes

If Health Center detects any problems with the object repository database connection, Health Center instructs you to launch the Narrowcast Server Administrator.

Path Environment Variable Validation

This system check validates that the path environment variable includes the location of the MicroStrategy Common Files so that Narrowcast Server can access those files.

This system check is available for Windows machines.

If the path environment variable does not include the location of the common files, you can add the common files to the path by clicking the link provided.

System checks for MicroStrategy Web products

Available Memory for JVM setting

This system check verifies that the machine hosting the MicroStrategy Web server has enough memory available to start the Web server. It checks to see if the machine has enough physical memory to support the JVM Max Heap Size as specified in the registry.

Directories Permissions

This system check verifies that the network account used to run the application server for MicroStrategy Web can read from and write to the application server directories.

If that account does not have read and write permissions for the directories, this system check fails. To resolve this problem, the network administrator should grant read and write permissions for those directories to the account being used, or the system administrator should select a different location for the application server to use so that the account has read and write permissions.

© 2016, MicroStrategy Inc.

On the **Configuration** tab, specify a user name to activate the application server for MicroStrategy Web.

JVM Maximum Heap Size

This system check confirms that the Java Virtual Machine has enough memory available. It performs the following actions:

- Checks the value of the JVM maximum heap size in the registry to ensure that it is at least 512 MB.
- Checks the MicroStrategy Web log files for any Out Of Memory errors.

If the JVM maximum heap size is under 512 MB or if MicroStrategy Web has recently experienced any Out Of Memory errors, you can type a new value for the maximum heap size in the **JVM Max Heap Size** field.

On a Windows machine, you can choose to restart IIS when you change the JVM maximum heap size.

Temporary Folder Available Disk Space

This system check verifies that enough disk space is available for MicroStrategy Web to function properly. It checks the available disk space for the operating system's Temp folder.

If this folder has less space available than the minimum values specified on the **Configuration** tab, either in MB or as a percentage of total disk space, this system check fails. To resolve this problem, some files should be deleted from the machine hosting the folder, or the system administrator should choose a different storage location with more available space.

On the **Configuration** tab, specify the minimum free disk space that must be available for the Temp folder, either in MB or as a percentage of total disk space. The system check fails if either condition is not met. A value of -1 indicates no limit. The default values are 1024 MB and 10 percent of total disk space.

Using a response file to configure a Health Agent or Master Health Agent

As an alternative to stepping through each page of the Configuration Wizard during the configuration process, you can create a response file with the Health Center configuration information and use that response file with the Configuration Wizard to automatically configure Health Center on this machine.

 © 2016, MicroStrategy Inc.

Creating a response file

MicroStrategy recommends that you create a response file through the graphical interface of the Configuration Wizard. You step through the Configuration Wizard and make your selections, as described in *To configure a machine as a Master Health Agent, page 703* or *To configure a machine as a Health Agent, page 704*. When you reach the Summary page of the Configuration Wizard, do not click **Finish**. Instead, click **Save**. The wizard prompts you to save your selections in a response file.

You can also create or modify a response file with a text editor. For information on all the parameters in the response file, see *Health Center response file parameters, page 725*.

MicroStrategy supplies a blank response file template, `Response.ini`, in the Common Files folder of your MicroStrategy installation. By default, this folder is `C:\Program Files (x86)\Common Files\MicroStrategy`.

Executing a response file

You can execute a response file in any of the following ways:

- From within the Configuration Wizard. See *To use a response file with the Configuration Wizard, page 723*.
- From the Windows command line. See *To use a response file through the Windows command line, page 724*. This enables users to run the file without using any graphical user interfaces.
- In UNIX or Linux. See *To use a response file through the Configuration Wizard in UNIX or Linux, page 724* or *To use a response file through the UNIX/Linux command line, page 724*.

To use a response file with the Configuration Wizard

1. From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Configuration Wizard**. The Configuration Wizard opens.
2. Click **Load**. The Open dialog box displays.
3. Browse to the path where the response file is saved and click **Open**. The Summary page opens.
4. An overview of all the configuration tasks performed by the response file is displayed. Review the configuration tasks. To perform the configuration, click **Finish**.

© 2016, MicroStrategy Inc.

To use a response file through the Windows command line

1. Type the following command in the Windows command line:

   `macfgwiz.exe -r "Path\response.ini"`

   Where *Path* is the fully qualified path to the response file. For example, a common location of a response file is:

   `C:\Program Files (x86)\Common Files\MicroStrategy\RESPONSE.INI`

   If an error message is displayed, check the path and name you supplied for the response file and make any required changes.

To use a response file through the Configuration Wizard in UNIX or Linux

1. From a UNIX or Linux console window, browse to *<HOME_PATH>*, where *<HOME_PATH>* is the directory you specified as the Home Directory during installation.
2. Browse to the folder `bin`.
3. Type `mstrcfgwiz-editor` and press **ENTER**. The Configuration Wizard opens with the Welcome page displayed.
4. Press **ENTER**.
5. Type `1` to select to use a response file and press **ENTER**.
6. Type the fully qualified path to the `response.ini` file and press **ENTER**. For example:

   `/home/username/MicroStrategy/RESPONSE.INI`

   If an error message is displayed, check the path and name you supplied for the response file and make any required changes.

To use a response file through the UNIX/Linux command line

1. From a UNIX or Linux console window, browse to *<HOME_PATH>*, where *<HOME_PATH>* is the directory you specified as the Home Directory during installation.
2. Browse to the folder `bin`.
3. Type the following command in the command line and press **ENTER**.

   `mstrcfgwiz-editor -response /Path/response.ini`

 © 2016, MicroStrategy Inc.

Where *Path* is the fully qualified path to the response file. For example, a common location of a response file is:

`/home/username/MicroStrategy/RESPONSE.INI`

If an error message is displayed, check the path and name you supplied for the response file and make any required changes.

Health Center response file parameters

Health Agent/Master Health Agent configuration

The response file parameters in the `[HealthCenter]` section configure a machine as a Health Agent or Master Health Agent. The table below lists the available parameters and the functionality of options for each parameter. For detailed information about each parameter, see the Configuration Wizard Help.

Options	Description
`[HealthCenter]`	Options in this section refer to configuring this machine as a Health Agent or Master Health Agent.
`HealthCenter=`	Defines whether to configure Health Center, as determined by the following values: • `1`: Configure Health Center on this machine. • `0`: Do not configure Health Center.
`ConfigWiz-NodeType=`	Defines whether to configure this machine as a Health Agent or a Master Health Agent: • `Health Agent`: Configure this machine as a Health Agent. • `Master Health Agent`: Configure this machine as a Master Health Agent.
`ConfigWiz-InstallationPath=`	The path in which Health Center is installed.
`ConfigWiz-HomePath=`	The path to the main MicroStrategy directory. On a Windows machine, by default this is C:\Program Files (x86)\MicroStrategy\.
`ConfigWiz-CommonPath=`	The path to the MicroStrategy common files directory. On a Windows machine, by default this is C:\Program Files (x86)\Common Files\MicroStrategy\.
`ConfigWiz-NetworkAddress=`	The network name of the machine running Health Center.
`ConfigWiz-NetworkPort`	The network port used by Health Center.
`ConfigWiz-KIDetectors-WebUniversal-onoff=`	Defines whether MicroStrategy Web Universal is deployed on this machine without using the MicroStrategy installer.

© 2016, MicroStrategy Inc.

Options	Description
	• `on`: Web Universal has been deployed on this machine without using the MicroStrategy installer. • `off`: Web Universal has not been deployed on this machine without using the MicroStrategy installer.
`ConfigWiz-KIDetectors-WebServicesUniversal-onoff=`	Defines whether MicroStrategy Web Services Universal is deployed on this machine without using the MicroStrategy installation. • `on`: Web Services Universal has been deployed on this machine without using the MicroStrategy installer. • `off`: Web Services Universal has not been deployed on this machine without using the MicroStrategy installer
`ConfigWiz-Marconi=` (Master Health Agent only)	Whether to join the MicroStrategy Customer Experience Improvement Program: • `true`: Join the MicroStrategy Customer Experience Improvement Program. • `false`: Do not join the MicroStrategy Customer Experience Improvement Program.
`ConfigWiz-RepositoryPath=` (Master Health Agent only)	The path to the Health Center Repository.
`ConfigWiz-UpperInclusiveRetryInterval=` (Master Health Agent only)	240 (This parameter should be changed only as instructed by MicroStrategy technical support.)
`ConfigWiz-LowerInclusiveRetryInterval=` (Master Health Agent only)	1200 (This parameter should be changed only as instructed by MicroStrategy technical support.)
`ConfigWiz-ProxySettings-ProxyEnabled=` (Master Health Agent only)	Whether to use a proxy server with Health Center: • `true`: Use a proxy server. • `false`: Do not use a proxy server.
`ConfigWiz-ProxySettings-ProxyHostName=` (Master Health Agent only)	The network name of the machine to use as a proxy server.
`ConfigWiz-ProxySettings-ProxyPort=` (Master Health Agent only)	The network port to use on the proxy server.

 © 2016, MicroStrategy Inc.

Options	Description
`ConfigWiz-ProxySettings-ProxyAuthEnabled=` (Master Health Agent only)	Whether to use authentication with the Health Center proxy server: • `true`: Use authentication. • `false`: Do not use authentication.
`ConfigWiz-ProxySettings-ProxyUserID=` (Master Health Agent only)	The username to log in to the proxy server.
`ConfigWiz-ProxySettings-ProxyPWD=` (Master Health Agent only)	A hash of the password for the username above.
`ConfigWiz-MachineCredentials-Mode=` (Health Agent only)	Defines whether the Health Agent access code is encrypted in the response file, as determined by the following values: • `0`: The access code is not encrypted in the response file, which enables you to modify the access code in the response file later using a text editor. You can then distribute the response file to multiple users with various login and password credentials. However, be aware that this can compromise your database security if you do not remove the access code from the response file before distributing it. • `1`: Encrypts the access code in the response file, which ensures that your access code is secure. This is the default behavior.
`ConfigWiz-MachineCredentials-Password=` (Health Agent only)	The Health Agent access code. This may be encrypted, depending on the value in the `ConfigWiz-MachineCredentials-Mode=` parameter.
`ConfigWiz-MachineCredentials-TrustedIPs-TrustedIP1..n=` (Health Agent only)	A trusted IP address for connecting to Health Center. To make multiple IP addresses available, include additional `ConfigWiz-MachineCredentials-TrustedIPs-TrustedIP`*n* parameters, and increase the value of N (the last character of the parameter name) for each different trusted IP address.

Health Agent connectivity

The response file parameters in the `[HealthCenterConnectivity]` section refer to options used to connect a local Health Agent to a Health Center system. The table below lists the parameters and the functionality of options for each parameter. For detailed information about each parameter, see the Configuration Wizard Help.

Options	Description
`[HealthCenterConnectivity]`	Options in this section refer to connecting a local Health Agent to a Health Center System.
`HealthCenterConnectivity=`	Defines whether to configure Health Center, as determined by the following values: • `1`: Configure Health Center on this machine • `0`: Do not configure Health Center
`UpConnections-UpConnection-Address=`	If this Health Agent is connecting to the Master Health Agent through another Health Agent, the machine name or IP address of that Health Agent machine.
`UpConnections-UpConnection-Port=`	The port for the Health Agent above.
`UpConnections-UpConnection-MHAAddress`	The machine name or IP address of the Master Health Agent machine.
`UpConnections-UpConnection-MHAPort`	The port for the Master Health Agent.
`UpConnections-UpConnection-MHAPassword=`	The access code for the Master Health Agent.

 © 2016, MicroStrategy Inc.

18

TROUBLESHOOTING

Introduction

This chapter provides guidance for finding and fixing trouble spots in the system. While the chapter does not go into great detail, it does provide references to the relevant portions of this guide where the topic or remedy is discussed in more detail. This chapter contains the following information:

- *Methodology for finding trouble spots, page 730*
- *Finding trouble spots using diagnostics, page 731*
- *Memory depletion troubleshooting, page 750*
- *Authentication troubleshooting, page 756*
- *Fixing inconsistencies in the metadata, page 762*
- *Object dependencies troubleshooting, page 765*
- *Date/time functions troubleshooting, page 765*
- *Performance troubleshooting, page 766*
- *Troubleshooting data imported from a file, page 767*
- *Subscription and report results troubleshooting, page 768*
- *Internationalization troubleshooting, page 771*
- *Troubleshooting Intelligence Server, page 772*
- *Clustered environments troubleshooting, page 774*

© 2016, MicroStrategy Inc.

- *Statistics logging troubleshooting, page 777*
- *Additional resources for troubleshooting, page 780*

Methodology for finding trouble spots

This section discusses how you can determine where the bottlenecks, breaking points, or other problem areas are in the MicroStrategy business intelligence system.

Locate a problem in the following list to find a suggested resolution.

Intelligence Server stopped

- Use MicroStrategy diagnostics logs to discover what happened (see *Finding trouble spots using diagnostics, page 731*)
- Use the server state dump (the `DSSErrors` log file) (see *Analyzing a server state dump, page 747*) to determine whether it was a:
  - Memory depletion (see *Memory depletion troubleshooting, page 750*)
  - Exception error (see *Working with exceptions, page 747*)
- Tune the system as necessary (see *Chapter 8, Tuning Your System for Best Performance*)

The system is slow

- Use reports in MicroStrategy Enterprise Manager (see the *MicroStrategy Operations Manager Guide*) to see:
  - Which components of the system are slow (use the "Execution Cycle Breakdown" report) to see if reports can be designed differently
  - When the system is slowest and if that relates to concurrency (use the "Peak Time Period," "Average Execution Time vs. Number of Sessions," and "Average Execution Time vs. Number of Jobs per User" reports)
  - Whether scheduled reports are running during peak times (use the "Scheduled Report Load" report) and if so, schedule them at off-peak times
  - Whether caching certain reports would improve performance (use the "Cache Analysis," "Top 10 Reports," and "Top 10 Longest Executing" reports)
- Tune the system (see *Chapter 8, Tuning Your System for Best Performance*).

 © 2016, MicroStrategy Inc.

Users cannot log in

- See *Authentication troubleshooting, page 756*

Intelligence Server does not start

- The connection to metadata may not be working (see *Connecting to the MicroStrategy metadata, page 7*)
- Or see *Troubleshooting Intelligence Server, page 772*.

Intelligence Server does not run reports

- The connection to the data warehouse may not be working or there may be problems with the data warehouse (see *Connecting to the data warehouse, page 8*)
- The result set row for the report may have exceeded the limit specified in the Project Configuration Editor or the VLDB Properties editor. (see *Subscription and report results troubleshooting, page 768*)

MicroStrategy Web or Web Universal cannot connect to Intelligence Server

- Check that the correct port numbers are set if you are using firewalls in your configuration (see *Using firewalls, page 534*)

Finding trouble spots using diagnostics

In MicroStrategy, the process of logging and analyzing operation and performance information is known as *diagnostics*. Information can be logged for many Intelligence Server and operating system features and functions. You can configure the log files to record diagnostics information at different levels to different files. For example, you can log all MicroStrategy errors to the default log file of `DSSErrors.log`, and log all information about Memory Contract Manager (see *Governing Intelligence Server memory use with Memory Contract Manager, page 316*) to a new log file called `MemoryLog.log`. You can also log performance information, such as the time taken to perform various operations and the total number of operations performed.

However, if too much information is logged, it can degrade the system's performance. By default, logging is set to a minimum. At some point you may want to detect problems in the system for which logging is not enabled by default.

This section includes information on the following topics:

- *Configuring what is logged, page 732*
- *Creating and managing log files, page 743*
- *Analyzing a server state dump, page 747*

Configuring what is logged

The MicroStrategy Diagnostics and Performance Logging tool configures which diagnostic messages are recorded to MicroStrategy log files. You can customize the logging options to gather information from more or fewer system components and performance counters, and to save log messages to different log files.

To configure logging with the Diagnostics and Performance Logging Tool

If you save any changes to settings in the Diagnostics and Performance Logging tool, you cannot automatically return to the out-of-the-box settings. If you might want to return to the original default settings at any time, record the default setup for your records.

1 Open the Diagnostics and Performance Logging Tool.

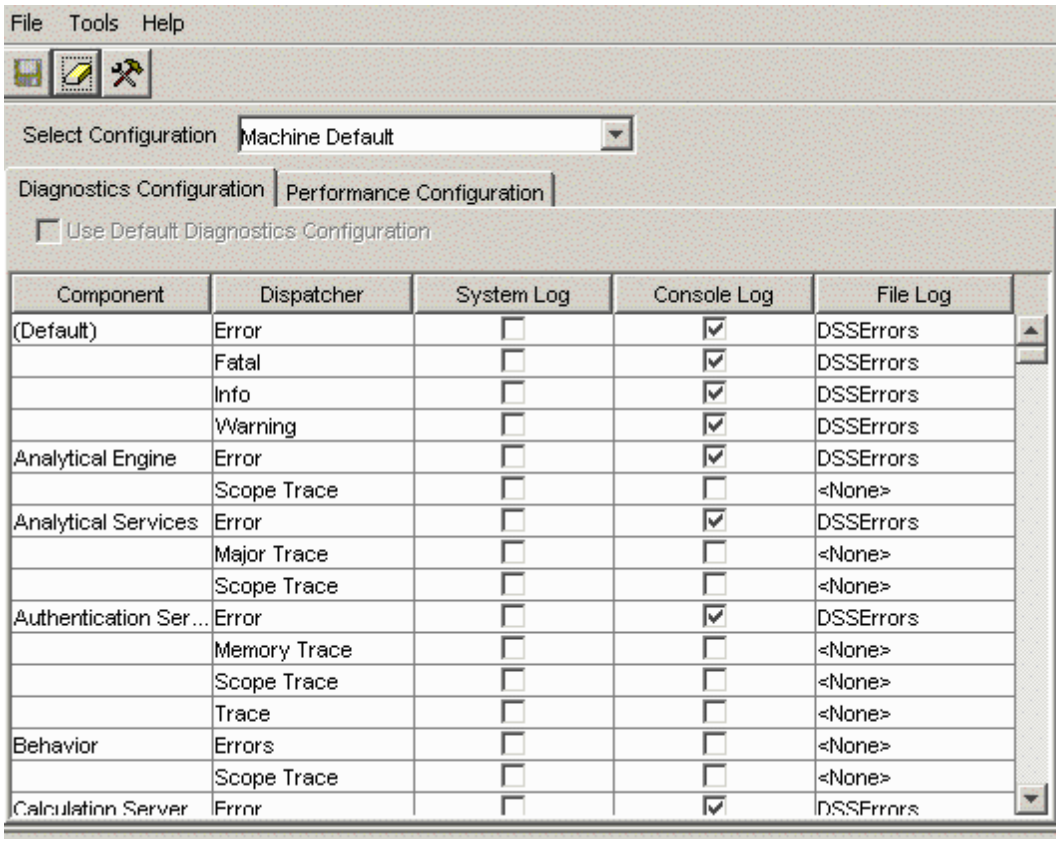
- From Developer: From the **Tools** menu, select **Diagnostics**.

If the **Diagnostics** option does not appear on the **Tools** menu, it has not been enabled. To enable this option, from the **Tools** menu, select **MicroStrategy Developer Preferences**. In the **General** category, in the **Advanced** subcategory, select the **Show Diagnostics Menu Option** check box and click **OK**.

- In Windows: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Diagnostics Configuration**.
- In UNIX/Linux: Navigate to the directory `~/MicroStrategy/bin` and enter `mstrdiag`.

The Diagnostics and Performance Logging Tool opens.

 © 2016, MicroStrategy Inc.



2 From the **Select Configuration** drop-down list, select whether to configure logging for this machine only or for the entire server instance:

- To configure logging for this machine only, select **Machine Default**.
- To configure logging for the server instance, select **CastorServer Instance**.

To configure the server instance with the logging settings that are used by this machine, select **CastorServer Instance** and then select the **Use Machine Default Diagnostics Configuration** check box.

Configure diagnostics logging

3 Select the **Diagnostics Configuration** tab. For more information about diagnostics logging, see *Diagnostics configuration, page 734*.

4 To log information about a component to the operating system log file, select the **System Log** check box for that component.

5 To log information about a component to the MicroStrategy Monitor console, select the **Console Log** check box for that component.

This log destination is intended for use for interactive testing and troubleshooting purposes, and should not be used in production deployments.

6 To log information about a component to a MicroStrategy log file, in the **File Log** drop-down list for that component, select the log file.

Logging the Kernel XML API component can cause the log file to grow very large. If you enable this diagnostic, make sure the log file you select in the **File Log** column has a **Max File Size (KB)** of at least 2000. For instructions on how to set the maximum size of a log file, see *Creating and managing log files, page 743*.

Configure performance logging

7 Select the **Performance Configuration** tab. For more information about performance logging, see *Configure performance logging settings, page 736*.

8 Configure the performance log file and statistics logging properties using the options on the right side of the Diagnostics and Performance Logging Tool. For detailed information about these options, click **Help**.

9 To log information from a performance counter to the performance log file, select the **File Log** check box for that counter.

10 To log information from a performance counter to the statistics tables, select the **Statistics** check box for that counter.

Save your changes

11 From the **File** menu, select **Save**. Your new settings are saved in the registry, and Intelligence Server begins logging the information that you configured.

You may need to restart Intelligence Server for the new logging settings to take effect.

Once the system begins logging information, you can analyze it by viewing the appropriate log file. For instructions on how to read a MicroStrategy log file, see *Creating and managing log files, page 743*.

Diagnostics configuration

Each component of the MicroStrategy system can produce log messages. These messages can help you track down the source of any errors that you encounter. For example, if your system seems to be running low on memory, you can view the log files to determine which components and processes are using more memory than anticipated.

These log messages can be recorded in a MicroStrategy log file. They can also be recorded in the operating system's log file, such as the Windows Event Monitor.

The component/dispatcher combinations that you choose to enable logging for depend on your environment, your system, and your users' activities. In general, the most useful dispatchers to select are the following:

 © 2016, MicroStrategy Inc.

- **Error**: This dispatcher logs the final message before an error occurs, which can be important information to help detect the system component and action that caused or preceded the error.
- **Fatal**: This dispatcher logs the final message before a fatal error occurs, which can be important information to help detect the system component and action that caused or preceded the server fatality.
- **Info**: This dispatcher logs every operation and manipulation that occurs on the system.

Some of the most common customizations to the default diagnostics setup are shown in the following table. Each component/dispatcher combination in the table is commonly added to provide diagnostic information about that component and its related trace (dispatcher). To add a combination, select its check box.

Component	Dispatcher
Authentication Server	Trace
Database Classes	All
Metadata Server	• Content Source Trace • Transaction Trace
Engine	DFC Engine
Element Server	• Element Source Trace • Object Source Trace
Object Server	• Content Source Trace • Object Source Trace • Scope Trace
Report Net Server	Scope Trace
Report Server	• Cache Trace • Object Source Trace • Report Source Trace
Kernel	• Scheduler Trace • User Trace
Kernel XML API	Trace **Note**: If you enable this diagnostic, make sure that the log file that you select in the **File Log** column has its **Max File Size (KB)** set to at least 2000.

Performance configuration

MicroStrategy components can also record various performance measurements. You can use these measurements to help tune your system for better performance, or to identify areas where performance can be improved. For example, you may want to discover exactly how much the CPU is used to perform a given system function.

Some performance counters can be logged to the Intelligence Server statistics tables as well. For more information about Intelligence Server statistics, see *Monitoring system usage: Intelligence Server statistics, page 288*.

Configure performance logging settings

When you select the performance counters to be recorded, you can determine how often data is recorded, and whether to persist the counters.

You can enable or disable performance logging without having to clear all the logging settings. To enable logging to a file, make sure the **Log Counters** drop-down list is set to **Yes**. To enable logging to the statistics database, make sure the **Persist Statistics** drop-down list is set to **Yes**.

To configure the performance logging settings

1. In the Diagnostics and Performance Logging tool, select the **Performance Configuration** tab.

Configure file logging options

2. From the **Log Destination** drop-down box, select the file to log performance counter data to.

   To create a new performance log file, from the **Log Destination** drop-down box, select **<New>**. The Log Destination Editor opens. For instructions on using the Log Destination Editor to create a new log file, click **Help**, or see *Creating and managing log files, page 743*.

3. In the **Logging Frequency (sec)** field, type how often, in seconds, that you want the file log to be updated with the latest performance counter information.

4. To log performance information to a log file, make sure the **Log Counters** drop-down list is set to **Yes**.

Configure statistics logging options

5. In the **Logging Frequency (min)** field, type how often, in minutes, that you want the statistics database to be updated with the latest performance counter information.

6. To log performance information to the statistics database, make sure the **Persist Statistics** drop-down list is set to **Yes**.

 © 2016, MicroStrategy Inc.

7 When you are finished configuring the performance counter log file, click **Save** on the toolbar. Your choices are saved for the selected log file.

Performance counters for specific MicroStrategy features

The table below lists the major MicroStrategy software features and the corresponding performance counters that you can use to monitor those features. For example, if the Attribute Creation Wizard seems to be running slowly, you can track its performance with the DSS AttributeCreationWizard, DSS ProgressIndicator, and DSS PropertySheetLib performance counters.

MicroStrategy Feature	Components	Trace Level
Attribute Creation Wizard	• DSS AttributeCreationWizard • DSS ProgressIndicator • DSS PropertySheetLib	Function Level Tracing
Attribute Editor	• DSS AttributeEditor • DSS ColumnEditor • DSS CommonDialogsLib • DSS Components • DSS EditorContainer • DSS EditorManager • DSS ExpressionboxLib • DSS FormCategoriesEditor • DSS PropertySheetLib	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.
Client Connection	• DSS AuthServer	• Authentication Tracing
	• DSS ClientConnection	• Session Tracing • Data Source Tracing • Data Source Enumerator Tracing
Consolidation Editor	• DSS CommonDialogsLib • DSS Components • DSS ConsolidationEditorLib • DSS EditorContainer • DSS EditorManager • DSS PromptsLib	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.

MicroStrategy Feature	Components	Trace Level
Custom Group Editor	• DSS CommonDialogsLib • DSS CommonEditorControlsLib • DSS Components • DSS DateLib • DSS EditorContainer • DSS EditorManager • DSS EditorSupportLib • DSS ExpressionboxLib • DSS FilterLib • DSS FTRContainerLib • DSS ObjectsSelectorLib • DSS PromptEditorsLib • DSS PromptsLib	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.
Data Transmitters and Transformers	• DSS DataTransmitter • DSS MhtTransformer • DSS MIME • DSS SMTPSender • DSS Network	• Function Level Tracing
Element Browsing	• DSS DBElementServer • DSS ElementNetClient • DSS ElementNetServer • DSS ElementServer	All components perform Element Source Tracing. DSS DBElementServer also performs Report Source Tracing.
Fact Creation Wizard	• DSS FactCreationWizard • DSS ProgressIndicator	Function Level Tracing
Fact Editor	• DSS ColumnEditor • DSS CommonDialogsLib • DSS Components • DSS EditorContainer • DSS EditorManager • DSS ExpressionboxLib • DSS ExtensionEditor • DSS FactEditor	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.
Filter Editor	• DSS CommonDialogsLib • DSS CommonEditorControlsLib	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.

 © 2016, MicroStrategy Inc.

MicroStrategy Feature	Components	Trace Level
	• DSS Components • DSS DateLib • DSS EditorContainer • DSS EditorManager • DSS EditorSupportLib • DSS ExpressionboxLib • DSS FilterLib • DSS FTRContainerLib • DSS ObjectsSelectorLib • DSS PromptEditorsLib • DSS PromptsLib	
Hierarchy Editor	• DSS CommonDialogsLib • DSS EditorContainer • DSS EditorManager • DSS HierarchyEditor	Function Level Tracing
HTML Document Editor	• DSS CommonDialogsLib • DSS Components • DSS DocumentEditor • DSS EditorContainer • DSS EditorManager	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.
Metadata SQL	• DSS MD4Server	• Object Tracing • Access Tracing
	• DSS MDServer	• SQL Tracing • Content Source Tracing
Metric Editor	• DSS CommonDialogsLib • DSS Components • DSS DimtyEditorLib • DSS EditorContainer • DSS EditorManager • DSS ExpressionboxLib • DSS MeasureEditorLib • DSS PromptsLib • DSS PropertiesControlsLib	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.
Object Browsing	• DSS ObjectServer • DSS SourceNetClient	All components perform Content Source Tracing. DSS ObjectServer also performs Object Source Tracing.

© 2016, MicroStrategy Inc.

MicroStrategy Feature	Components	Trace Level
	• DSS SourceNetServer	
Partition Editor	• DSS CommonDialogsLib • DSS Components • DSS DataSliceEditor • DSS EditorContainer • DSS EditorManager • DSS FilterLib • DSS PartitionEditor	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.
Print Schema	• DSS PrintCore • DSS PrintSchema • DSS ProgressIndicator	Function Level Tracing
Project Creation	• DSS AttributeCreationWizard • DSS FactCreationWizard • DSS ProgressIndicator • DSS ProjectCreationLib • DSS WHCatalog	Function Level Tracing
Project Duplication	• DSS AsynchLib • DSS ProgressIndicator • DSS ProjectUpgradeLib • DSS SchemaManipulation	Function Level Tracing
Project Upgrade	• DSS AsynchLib • DSS ProgressIndicator • DSS ProjectUpgradeLib • DSS SchemaManipulation	Function Level Tracing
Prompt Editor	• DSS CommonDialogsLib • DSS CommonEditorControlsLib • DSS Components • DSS EditorContainer • DSS EditorManager • DSS EditorSupportLib • DSS PromptEditorsLib • DSS PromptStyles • DSS SearchEditorLib	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.

 © 2016, MicroStrategy Inc.

MicroStrategy Feature	Components	Trace Level
Report Editor	• DSS CommonDialogsLib • DSS CommonEditorControlsLib • DSS Components • DSS DateLib • DSS EditorContainer • DSS EditorManager • DSS EditorSupportLib • DSS ExportLib • DSS ExpressionboxLib • DSS FilterLib • DSS FTRContainerLib • DSS GraphLib • DSS GridLib • DSS ObjectsSelectorLib • DSS PageByLib • DSS PrintGraphInterface • DSS PrintGridInterface • DSS PromptEditorsLib • DSS PromptsLib • DSS PropertySheetLib • DSS RepDrillingLib • DSS RepFormatsLib • DSS RepFormsLib • DSS ReportControl • DSS ReportDataOptionsLib • DSS ReportSortsLib • DSS ReportSubtotalLib	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.
Report Execution	• DSS ReportNetClient	• Report Source Tracing • Process Tracing
	• DSS ReportNetServer	• Process Tracing
	• DSS ReportServer	• Report Source Tracing

MicroStrategy Feature	Components	Trace Level
Server Administration	• DSS AdminEditorContainer • DSS DatabaseInstanceWizard • DSS DBConnectionConfiguration • DSS DBRoleConfiguration • DSS DiagnosticsConfiguration • DSS EVentsEditor • DSS PriorityMapEditor • DSS PrivilegesEditor • DSS ProjectConfiguration • DSS SecurityRoleEditor • DSS SecurityRoleViewer • DSS ServerConfiguration • DSS UserEditor • DSS VLDBEditor	Function Level Tracing
Table Editor	• DSS CommonDialogsLib • DSS EditorContainer • DSS EditorManager • DSS TableEditor	Function Level Tracing

 © 2016, MicroStrategy Inc.

MicroStrategy Feature	Components	Trace Level
Template Editor	• DSS CommonDialogsLib • DSS Components • DSS EditorContainer • DSS EditorManager • DSS ExportLib • DSS FTRContainerLib • DSS GraphLib • DSS GridLib • DSS PageByLib • DSS PrintGraphInterface • DSS PrintGridInterface • DSS PromptsLib • DSS PropertySheetLib • DSS RepDrillingLib • DSS RepFormatsLib • DSS RepFormsLib • DSS ReportControl • DSS ReportDataOptionsLib • DSS ReportSortsLib • DSS ReportSubtotalLib	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.
Transformation Editor	• DSS CommonDialogsLib • DSS Components • DSS EditorContainer • DSS EditorManager • DSS ExpressionboxLib • DSS TransformationEditor	All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.
Warehouse Catalog Browser	• DSS CommonDialogsLib • DSS DatabaseInstanceWizard • DSS DBRoleConfiguration • DSS SchemaManipulation • DSS WHCatalog	Function Level Tracing

Creating and managing log files

Diagnostics information can be logged to multiple log files. For example, in the default configuration, all error messages are logged to `DSSErrors.log`, license information is

logged to `License.log`, and messages from the Java Virtual Machine in MicroStrategy Web are logged to `JVMMessages.log`.

Performance information must all be logged to the same log file.

Each log file has a specified maximum size. When a MicroStrategy log file reaches its maximum size, the file is renamed with a `.bak` extension, and a new log file is created using the same file name. For example, if the `DSSErrors.log` file reaches its maximum size, it is renamed `DSSErrors.bak`, and a new `DSSErrors.log` file is created.

You can create new log files and change the maximum size of log files in the Log Destination Editor.

To change the maximum size of a log file

1. In the Diagnostics and Performance Logging Tool, from the **Tools** menu, select **Log Destinations**. The Log Destination Editor opens.
2. From the **Select Log Destination** drop-down list, select the log file.
3. In the **Max File Size (KB)** field, type the new maximum size of the log file, in kilobytes.

   If the Kernel XML API component is selected in the **Diagnostics** tab, the **Max File Size** for that file should be set to no lower than 2000 KB.
4. Click **Save**. The change is saved.
5. Click **Close**. The Log Destination Editor closes.

To create a new log file

1. In the Diagnostics and Performance Logging Tool, from the **Tools** menu, select **Log Destinations**. The Log Destination Editor opens.
2. From the **Select Log Destination** drop-down list, select **<New>**.
3. In the **File Name** field, type the name of the file. The `.log` extension is automatically appended to this file name.
4. In the **Max File Size (KB)** field, type the maximum size of the new log file, in kilobytes.
5. Click **Save**. The new log file is created and available for use.
6. Click **Close**. The Log Destination Editor closes.

 © 2016, MicroStrategy Inc.

Viewing and analyzing log files

All MicroStrategy log files are stored in the log file location. This location is set during installation and cannot be changed.

- On Windows, all log files are stored in `C:\Program Files (x86)\Common Files\MicroStrategy\Log\`.
- On Unix:
  - Log files for the CastorServer Instance are stored in `home/USER/MicroStrategy/log/Intelligence Server`.
  - Log files for the Machine Default are stored in `home/USER/MicroStrategy/log`.

These log files are plain text files and can be viewed with any text editor. For a description of how to read a log file, see *Anatomy of a log file, page 745*.

If your machine is configured as a MicroStrategy Health Center Health Agent or Master Health Agent, you can view log files through the Health Center Console. You can also filter the log files. For example, you can view entries relating only to memory conditions if you are trying to troubleshoot a memory depletion. For more information about viewing log files in Health Center, see *Chapter 17, Maintaining Your MicroStrategy System with Health Center*.

The MicroStrategy Web server error log files are in the `MstrWeb/WEB-INF/log/` directory. These log files can be viewed from the Web Administrator page, by clicking **View Error log** on the left side of the page. (In Web Universal, the link is **View logs**.) For more information about viewing log files in MicroStrategy Web, see the Web Administrator Help (from the Web Administrator page, click **Help**).

Anatomy of a log file

All messages in the log files have the same format. Each entry has the following parts:

```
PID:[thread][date::time][module name][trace type]message
```

Section	Definition
PID	Numeric ID of the process that performed the action
thread	Numeric ID of the thread that performed the action
date::time	Date and time at which the action happened
module name	Name of the MicroStrategy component that performed the action
trace type	Type of the log file entry
message	Message about the action

Sample log file

The following sample is a simple log file that was generated from MicroStrategy Web (ASP.NET) after running the report called Length of Employment in the MicroStrategy Tutorial project. The bulleted line before each entry explains what the log entry is recording.

• Intelligence Server creates a report definition.
`286:[THR:480][02/07/2003::12:24:23:860][DSS ReportServer][Report Source Tracing]Creating Report(Definition) with Flags=0x1000180 (OSrcCch UptOSrcCch)`
• Intelligence Server loads the report definition object named Length of Employment from the metadata.
`286:[THR:480][02/07/2003::12:24:23:860][DSS ReportServer][Report Source Tracing] where Definition = Object(Name="Length of Employment" Type=3(Report Definition) ID=D1AE564911D5C4D04C200E8820504F4F Proj=B19DEDCC11D4E0EFC000EB9495D0F44F Ver=493C8E3447909F1FBF75C48E11AB7DEB)`
• Intelligence Server creates a report instance named Length of Employment.
`286:[THR:480][02/07/2003::12:24:24:931][DSS ReportServer][Report Source Tracing]Created ReportInstance(Name="Length of Employment" ExecFlags=0x1000180(OSrcCch UptOSrcCch) ExecActn=0x1000180(RslvCB LclCch))`
• Intelligence Server begins executing the report instance.
`286:[THR:480][02/07/2003::12:24:24:931][DSS ReportServer][Report Source Tracing]Executing ReportInstance(Name="Length of Employment" ExecFlags=0x1000180(OSrcCch UptOSrcCch) ExecActn=0x1000180(RslvCB LclCch)) with Actions=0x8300003f(Rslv GenSQL ExeSQL Alrt XTab EvalVw LclCch UptLclCch), Flags=0x1000180(OSrcCch UptOSrcCch)`
• Intelligence Server checks to see whether the report exists in the report cache.
`286:[THR:480][02/07/2003::12:24:25:181][DSS ReportServer][Report Source Tracing]Finding in cache: ReportInstance(Name="Length of Employment" ExecFlags=0x1000180(OSrcCch UptOSrcCch) ExecActn=0x1000180(RslvCB LclCch))`
• Intelligence Server did not find the report in the cache.
`286:[THR:480][02/07/2003::12:24:25:342][DSS ReportServer][Report Source Tracing]Not found in cache: ReportInstance(Name="Length of Employment" ExecFlags=0x1000180(OSrcCch UptOSrcCch) ExecActn=0x1000180(RslvCB LclCch))`
• Intelligence Server checks for prompts and finds none in the report.

 © 2016, MicroStrategy Inc.

```
286:[THR:314][02/07/2003::12:24:25:432][DSS ReportServer][Report
Source Tracing]No prompts in ReportInstance(Name="Length of
Employment" ExecFlags=0x1000180(OSrcCch UptOSrcCch)
ExecActn=0x1000180(RslvCB LclCch))
```

- Intelligence Server executes the report and updates the caches.

```
286:[THR:492][02/07/2003::12:24:26:634][DSS ReportServer][Report
Source Tracing]Executing ReportInstance(Job=2 Name="Length of
Employment" ExecFlags=0x1000184(OSrcCch UptOSrcCch)
ExecActn=0x1000184(ExeSQL RslvCB LclCch)) with Actions=0x300003f(Rslv
GenSQL ExeSQL Alrt XTab EvalVw LclCch UptLclCch), Flags=0x1000184
(OSrcCch UptOSrcCch)
```

More detail is logged for report execution if the report is run from Developer.

Working with exceptions

When Intelligence Server encounters an error, it "throws an exception." Not all exceptions are fatal; in fact Intelligence Server uses some of them internally. Fatal exceptions cause Intelligence Server to shut down and they are logged in the `DSSErrors.log`, often as "unknown exceptions."

Fatal exception messages by themselves are not sufficient for accurate diagnosis. Intelligence Server includes a built-in mechanism to capture structured exceptions and generate a dump file that has more information in it. You may need to do this for MicroStrategy Technical Support specialist. For more information on diagnosing exceptions in this manner, see MicroStrategy Tech Note TN13130.

Analyzing a server state dump

A *server state dump (SSD)* is a collection of information related to the state of Intelligence Server that is written to the `DSSErrors.log` file, usually as a result of an unexpected shutdown of Intelligence Server. It provides insight into what was going on in Intelligence Server when the shutdown occurred. This information can be used to help diagnose the cause of the shutdown and avert subsequent problems.

Problems that trigger an SSD include memory depletion (see *Memory depletion troubleshooting, page 750*) or exceptions (see *Working with exceptions, page 747*). Changes to the server definition trigger a subset of the SSD information.

Analyzing a server state dump

Each SSD records information under the same process ID and thread ID. This information includes the server and project configuration settings, memory usage, schedule requests, user sessions, executing jobs and processing unit states, and so on. The SSD information is broken into 14 sections, summarized below.

Section 1: Triggering error and error specific preamble

This section precedes the actual SSD and provides information on what triggered the SSD, such as memory depletion or an unknown exception error.

Section 2: Server executable version and build information

This section provides information on the Intelligence Server executable version and build time so you can accurately identify the version of the MicroStrategy software.

Section 3: Server definition basic (Castor Server Configuration 'project') information

This section provides a subset of Intelligence Server level settings as they are defined in the Intelligence Server Configuration Editor (in Developer, right-click the project source, and select **Configure MicroStrategy Intelligence Server**). The settings consist of:

- Server definition name
- Maximum jobs per project
- Maximum connections per project
- Number of projects
- Communication protocol and port

WorkingSet File Directory and Max RAM for WorkingSet Cache values are not listed in an SSD.

Section 4: Project/s basic information

This section includes basic information related to the state and configuration of projects. This shows settings that are defined in the Project Configuration Editor, such as:

- Project name
- Cache settings
- Governor settings
- DBRole used
- DBConnection settings

Section 5: Server definition advanced information

This section includes additional server definition settings, such as:

- Thread load balancing mode
- Memory throttling

 © 2016, MicroStrategy Inc.

- History List settings
- Idle timeouts
- XML governors
- Memory Contract Manager (MCM) settings

MCM is designed to help you avoid memory depletions. For more information on MCM, see *Governing Intelligence Server memory use with Memory Contract Manager, page 316*.

Section 6: Callstack, lockstack, and loaded modules

The callstack dump provides information on the functions being used at the time the SSD was written. Similarly, the lockstack provides a list of active locks. The Module info dump provides a list of files that are loaded into memory by Intelligence Server, and their location in memory.

This information can help MicroStrategy Technical Support trace errors to specific areas of functionality.

Section 7: Server process memory snapshot

This section contains the memory profile of the Intelligence Server process and machine. If any of these values are near their limit, memory may be a cause of the problem.

Section 8: Project state summary

This section provides a summary of whether each project is Loaded and Registered, and the number of users logged in and jobs running at the time of the SSD.

Section 9: Schedule request information

This section provides a listing of schedule requests that Intelligence Server is configured for. This list includes:

- Reports
- Documents
- Administration tasks, such as idling projects and other tasks related to cache management

For additional information about schedules and subscriptions, see *Chapter 12, Scheduling Jobs and Administrative Tasks*.

© 2016, MicroStrategy Inc.

Section 10: Database connection snapshot

This section displays a snapshot of the state of the database connections between Intelligence Server and the metadata and data warehouse databases. This information is similar to what is shown in the Database Connection Monitor. For more information about database connections, see *Communicating with databases, page 6*.

Section 11: User Inbox snapshot

The section provides information on the size of various user inboxes and information related to the WorkingSet.

Section 12: Jobs status snapshot

This section provides a snapshot of the jobs that were executing at the time of the SSD. This information may be useful to see what the load on Intelligence Server was, as well as what was executing at the time of the error. If the error is due to a specific report, the information here can help you reproduce it.

Section 13: User session snapshot

This section provides details on the various user sessions in Intelligence Server at the time of the SSD.

Section 14: Processing Unit threads state snapshot

This section provides information about the states of the threads in each processing unit in Intelligence Server. It also provides information on the number of threads per Processing Unit and to what priority they are assigned.

Memory depletion troubleshooting

Memory depletion in Intelligence Server is a common cause of problems in the MicroStrategy system. Knowing how to prevent them enables you to make your system more stable and reliable. Intelligence Server's memory use is discussed earlier in this guide (see *Memory, page 312*). This memory troubleshooting material is merely a recap of the information presented there. In short, here is how you can limit memory depletions:

- Enable the Memory Contract Manager. This is a built in tool that controls whether certain job tasks are allowed to occur according to how much memory they could consume. This does not guarantee that a memory depletion will not occur, but it decreases the chance of it. For more information on this, see *Governing Intelligence Server memory use with Memory Contract Manager, page 316*.

 © 2016, MicroStrategy Inc.

- Be proactive. During the project planning/building phase, apply those concepts that are presented in the Tuning section. For information, see *Chapter 8, Tuning Your System for Best Performance*.
- Be aware of the primary memory consumers and implement governors and system limits to reign them in.
- Monitor the system with the Windows Performance Monitor (see *Managing system memory and resources: Windows Performance Monitor, page 311*) and MicroStrategy Enterprise Manager (see the Enterprise Manager chapter in the *MicroStrategy Operations Manager Guide*). Users' practices of system use can evolve over time. New problems can arise as users are added to system and as they become savvy in using the system. These new potential problems may require tuning changes (see *Chapter 8, Tuning Your System for Best Performance*).

Basic memory concepts

This is a very brief summary of the memory basics. For more detail on this, see *Memory, page 312*.

Virtual memory is Physical memory (RAM) + Disk Page file (also called the swap file). It is shared by all processes running on the machine including the operating system.

The *user address space* (UAS) is independent of virtual memory and is of finite size. It is measured per process on the machine (such as the MSTRSVR.exe Intelligence Server application). By definition, in a 32-bit operating system, virtual bytes is limited to 4GB (2^{32}). The 32-bit Windows operating system divides this into two parts UAS and System Address Space (SAS). The UAS is, in this case, for Intelligence Server to store data and code, and the SAS is for the operating system's use.

Virtual bytes. measures the use of the UAS. When virtual bytes reaches the UAS limit, it causes a memory depletion.

The **Commit Limit** in Windows Task Manager is not equal to Virtual bytes.

Private bytes reflect virtual memory usage, and they are a subset of allocated virtual bytes.

To help determine what is causing memory depletion, answer these questions:

- How is the project being used? Are there very complex reports? Reporting on very large data sets?
- Is the Scheduler being used heavily? Are many reports running on a single schedule? Is there a schedule that runs during the peak usage time?
- What is the prompted/non-prompted report mix?
- How is the History List used? Are there many messages?
- Is there high user concurrency?
- Is there high job concurrency (either jobs or large reports)?
- Are the governor settings too high for working set, XML cells, result set, and so on?

© 2016, MicroStrategy Inc.

To answer these questions, you must be familiar with the system and how it is being used (Enterprise Manager reports will help you with this). But perhaps most useful is to know what the system was doing when the memory depletion occurred. To answer this question, use:

- The Windows Performance Monitor (see *Monitoring memory use with Performance Monitor, page 313*) to characterize memory use over time. Examine it in relation to job and user concurrency. Typically, log these counters:
  - Process / Virtual Bytes (MSTRSVR process)
  - Process / Private Bytes (MSTRSVR process)
  - Process / Thread Count (MSTRSVR process)
  - MicroStrategy Server Jobs / Executing Reports
  - MicroStrategy Server Users / Open Sessions

  Are there spikes in memory? Large reports or exports being executed? Is there an upward trend over time?
- The `DssErrors.log` file for details about what was happening when the memory depletion occurred (see *Creating and managing log files, page 743*)
- Your knowledge of the system and whether the top memory consumers are heavily used

Potential causes of memory depletion

Tasks and objects from the following categories are likely to consume the most memory in Intelligence Server:

- *Intelligence Server memory footprint, page 752*
- *Report cache memory, page 753*
- *Working set, page 753*
- *History List (Inbox) filled with messages, page 754*
- *A single large report, page 754*
- *Export to Excel from MicroStrategy Web or Web Universal, page 755*
- *Multiple memory-intensive requests on Linux, page 755*

Intelligence Server memory footprint

When Intelligence Server starts, several operations occur, all of which use memory. The following sections provide details:

- *What happens when Intelligence Server starts?, page 12*
- *How much memory does Intelligence Server use when it starts up?, page 314*

 © 2016, MicroStrategy Inc.

- *How does Intelligence Server use memory after it is running?, page 315*

Common factors for high memory usage on startup include:

- Many data warehouse connection threads (approximately 1MB/thread)
- The number of caches and their size
- Number of projects
- Project schema size
- Number of schedules (with autoprompts)

Possible solutions:

- Reduce the number of data warehouse connection threads or limit how long they can exist (see *Managing database connection threads, page 337*)
- Reduce the Maximum RAM size for report caches (see *Maximum RAM usage, page 421*, and below)
- Split projects to multiple Intelligence Servers

Report cache memory

The cache lookup table is responsible for most memory problems related to caching. It matches report requests to report caches and is loaded into memory on each cluster node. Prompt answers tend to consume most memory. To give you an idea of the size of the lookup table, calculate the size of all `CachLKUP.idx` files. Also, the cache lookup table memory consumption is not governed.

For more detailed information about report caches, see *Result caches, page 400*.

Possible solutions:

- Disable report caching
  - Is caching required? If most of your users do ad hoc reporting or if the data warehouse is updated frequently, caching may not be helping the efficiency of your system's operation.
  - Check the cache hit ratio using the Enterprise Manager report "Cache Analysis."
  - Heavy prompt usage decreases cache usage (disable caching of prompted reports).
- Reduce the Maximum RAM size for report caches (see *Maximum RAM usage, page 421*)

Working set

This feature's memory use typically correlates with the number of open user sessions and the size of reports the users run.

Possible solutions:

- Governing user resources (*Governing user resources, page 327*)
- Reduce the working set size
- Reduce size of reports
- Limit the user sessions either in number or by time (*Governing concurrent users, page 325*)

History List (Inbox) filled with messages

Many messages in the History List consume a lot of memory. When the user logs in to the system, his or her Inbox is loaded into memory. Every user logged in consumes more memory.

Possible solutions:

- Decrease the session idle timeouts so that the user's Inbox can be unloaded (see *Governing concurrent users, page 325*)
- Limit the number of Inbox Maximum number of messages per user (see *Governing user resources, page 327*).

A single large report

You may notice that when a certain report is run, memory use spikes. This could be caused by these factors:

- A large amount of data returned from the data warehouse
- Large amount of data retrieved for element prompts
- Multiple page-by fields and subtotals being used in the report
- Use of Custom groups
- High Analytical Engine complexity
- A large amount of XML returned to the MicroStrategy Web product
- Large reports in a document

Possible solutions:

For details on all the solutions below, see *Results processing, page 343* (unless otherwise noted).

- Redefine the report or split the report into multiple smaller reports
- Restrict object prompt options
    - Do not allow creating a report with object prompts. These prompts allow the user to throw many objects on a report. For example, a report designer might allow the user to select from a list of many metrics and attributes, with no restriction on the number of objects to place on the report. This makes it easy

 © 2016, MicroStrategy Inc.

for a user to create such a large report that, when executed, results in a large data set or complex set of multiple metrics that use lots of memory.

(To limit the number of prompt answers: When creating a prompt using the Prompt Generation Wizard, select the **Choose default prompt answers** check box, then click **Next**. You can then set a minimum and maximum number of prompt answers allowed.)

Another way to restrict prompt options is for the report designers to be more conscious about what they are doing when they give the user a list of prompt options.

- Restrict element prompts
  - For example, do not allow an element prompt to bring back a list of thousands of elements to the user (see *Limiting the number of elements displayed and cached at a time, page 443*). The process of getting all these elements back from the data warehouse can cause a memory depletion, especially if several of these are executing concurrently.
- Restrict the report's maximum size so users cannot create large reports (see *Governing results delivery, page 347*)
  - Reduce the Maximum number of XML cells setting
  - Reduce the Maximum number of rows setting

Export to Excel from MicroStrategy Web or Web Universal

Exporting a report to Excel (especially with formatting), uses a lot of memory because it returns all pages of the report at one time. You may see a type of "heartbeat" pattern in memory use because the XML is returned to Web in chunks. This chunk size = Maximum number of XML cells. For more information on this, see *What happens when I export a report from MicroStrategy Web?, page 44*.

Possible solutions:

- Have your users export using the "Plain text," "Excel with plain text," or "CSV file format" settings, which use less memory than others
- Reduce the Maximum number of XML cells setting (see *Governing results delivery, page 347*)

Multiple memory-intensive requests on Linux

A memory depletion may be reported on Linux even when the machine has plenty of free memory. This can be a result of the operating system reaching the maximum number of memory mapped blocks for the Intelligence Server process. The default maximum limit is 65536. To increase this limit, use the command

```
sysctl -w vm.max_map_count=655360
```

Authentication troubleshooting

This section gives you a list of things to check according to the type of authentication you are using:

- *Troubleshooting Windows authentication in MicroStrategy Web, page 756*
- *Failure to log in to server (three-tier) project source, page 756*
- *Troubleshooting LDAP authentication, page 757*

Troubleshooting Windows authentication in MicroStrategy Web

To use Windows authentication with MicroStrategy Web, you must be running Web or Web Universal on a Windows machine with Microsoft IIS. Non-IIS web servers do not support Windows authentication.

In cases where you have configured a project source to use Windows authentication, when logging in to MicroStrategy Web, your users may see the error message: "`You cannot login as an NT User. Please ask the MicroStrategy Server Administrator to link this NT User account to a MicroStrategy Server User.`" This may happen even if the user's Windows account is properly linked to the MicroStrategy user account. The problem may be that several settings in Internet Information Services (IIS) Service Manager are incorrect. To fix this, perform the following on each MicroStrategy Web server machine:

1. Access the IIS Internet Service Manager. The Microsoft Management Console dialog box opens.
2. Right-click the **MicroStrategy** virtual folder and select **Properties**.
3. On the **Directory Security** tab, under Anonymous Access and Authentication Control, click **Edit**. The Authentication Methods dialog box opens.
    - Clear the **Allow anonymous access** check box.
    - Select the **Windows Challenge/Response** check box.
4. Click **OK**, then **OK**.
5. Restart the machine for the changes to take effect.

Failure to log in to server (three-tier) project source

When logging into a 3-tier project source from Developer in Intelligence Server Universal, you might see the following error message: "`MsiSessionManager:failed`

 © 2016, MicroStrategy Inc.

`to login User administrator on machine x.x.x.x`". Additionally in the DSSErrors.log file, the following error message is displayed:

```
[Kernel][Error] ConfigManager::GetServerDefSetting():
ServerDef not initialized: Long SettingId=75.
```

This may happen because the server definition has not been initialized correctly.

To fix this, recreate the server definition as follows:

1. Launch the MicroStrategy Configuration Wizard. (In Windows, from the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Configuration Wizard**.)
2. In the Configuration Wizard, select **MicroStrategy Intelligence Server** and click **Next**. Proceed through the Configuration Wizard to create a new server definition.
3. Restart Intelligence Server to load the new server definition created above.
4. Connect to Intelligence Server from Developer via a 3-tier project source.

Troubleshooting LDAP authentication

LDAP authentication problems in Intelligence Server usually fall into one of these categories:

- Authentication issues that include clear text and Secure Socket Layer (SSL) connection modes
- Functionality problems/questions about importing users or groups, and synchronizing LDAP users in the MicroStrategy metadata

LDAP setup problems

The following list describes error messages that may in MicroStrategy when trying to connect to the LDAP Server.

Missing components: If you receive an error message stating that LDAP components could not be found, it may mean the DLL files are not in the appropriate directory. For details on choosing the correct DLL files according to your SDK and installing them in the correct location, see *Implementing LDAP authentication, page 105*. If you move your DLL files, be sure to restart Intelligence Server.

Invalid login/password: If you receive an "Incorrect login/password" error message, check your login name and password carefully. Also check that you have the correct DN search root and the correct user search filter syntax in the Intelligence Server Configuration Editor.

Cannot contact LDAP Server: If you receive a "Cannot connect to the LDAP Server" error message, try to connect using the LDAP Server IP address. You should also check to be sure the LDAP machine is running. Another possibility is that the SSL certificate files are not valid. For additional SSL troubleshooting, see the next section.

LDAP connection mode problems

The authentication process involves an authentication user which contacts the LDAP Server, and the actual user who is logging in. The authentication user is used by Intelligence Server to log in to the LDAP server and search for the actual user, using the actual user's Distinguished Name (DN). Therefore, the authentication user must have the necessary read and search privileges within the LDAP server and must be able to log in to a correct root. A correct root is characterized as one that contains each of the potential LDAP users who will be logging to Intelligence Server in one of its branches. Thus, if the authentication user cannot find the actual user in one of its branches, the search for the actual user fails.

There are two modes of connecting to your LDAP server: **Clear Text** and **SSL**. Depending on which you are using, answering the following questions may help you reach a solution:

- **If authentication fails in Clear Text mode**:
  - Can the authentication user log in as an LDAP user? The authentication user string can be tested in Developer using the **Test Connection** option in the Intelligence Server Configuration Editor. You can also test the user connection to the LDAP Server using any LDAP browser.

    - Is the authentication DN string correct?

    - Is the password for the authentication user correct?
  - Can the LDAP user (different from the authentication user) log in? You can test this with any LDAP browser.

    - Are the user credentials correct?

    - Do the LDAP server-side logs show success messages? The LDAP administrator can access these logs.
  - How can I make a Clear Text connection to Sun One/iPlanet on Solaris? For details and steps to connect to Sun One/iPlanet in Clear Text mode, refer to MicroStrategy Tech Note TN12066.
- **If authentication fails in SSL mode**:
  - Does the authentication work in clear text mode?
  - Are the LDAP-SDK dlls installed on the Intelligence Server machine?
  - Do the LDAP-SDK dlls reside in the correct system path?
  - Is the certificate obtained from the correct Certificate Authority (CA)? (For information on how to obtain the certificate from the corresponding CA platforms, search the MicroStrategy Knowledge Base for "LDAP AND certificate AND import.")
  - Does the certificate reside on the Intelligence Server machine in the correct system path?
  - Do the LDAP server side logs show success messages? The LDAP administrator can access these logs.

 © 2016, MicroStrategy Inc.

Authentication problems with LDAP attributes

You can integrate LDAP attributes in your MicroStrategy security model. For example, you can import the LDAP attribute `countryName`, and create a security filter that filters data by the user's country. Additionally, you can ensure that users cannot log in to projects if all required LDAP attributes are not defined, by enabling the **User login fails if LDAP attribute value is not read from the LDAP server** option.

If you select the option to prevent users from logging in if their LDAP attributes are not defined, be aware of the following potential issues:

- This setting prevents users from logging in to all projects in a project source.
- If you are using trusted authentication, such as with SiteMinder or Tivoli, and your trusted authentication provider is configured to use an LDAP directory, this option prevents trusted authentication users if they do not have all the required LDAP attributes.
- If your system has multiple LDAP servers, you must ensure that all the required LDAP attributes are defined on all LDAP servers. For example, if a required LDAP attribute is defined on LDAP server A, but not on LDAP server B, users from LDAP server B will not be able to log in to MicroStrategy.

LDAP functionality problems/questions

Functionality problems in MicroStrategy that are associated with LDAP authentication are most commonly caused by the integration of Intelligence Server with the LDAP servers. Once the authentication is successful (Intelligence Server has verified the existence of the LDAP user in the LDAP server), it needs to treat the LDAP user as a MicroStrategy user so that the user has the necessary privileges. (Note that privileges depend on how the LDAP user is authenticated. See *Implementing LDAP authentication, page 105* for more details.)

Intelligence Server achieves this transformation by importing the LDAP user as a new MicroStrategy user into the MicroStrategy metadata (the option not to import the LDAP user is discussed later). The relationship between the LDAP user and the MicroStrategy user is maintained using a link in the MicroStrategy metadata, which is in the form of a Distinguished Name (DN) specified for the user. A DN is the unique identifier of an entry in the LDAP directory.

You can choose to assign DNs to MicroStrategy users explicitly. If none is supplied, the LDAP user's DN is assigned to the MicroStrategy user after the LDAP user is imported. MicroStrategy uses the DN to locate users and groups in the LDAP Server even if LDAP users and groups are configured to be authenticated in MicroStrategy other than via import.

The MicroStrategy user's DN is different from the DN assigned for the authentication under LDAP configuration. The authentication user DN is the DN of the MicroStrategy account that logs in to the LDAP server and does the authentication (search/verification) for the actual user trying to log in. The authentication user can be anyone who has search privileges in the LDAP Server and is generally the LDAP administrator.

© 2016, MicroStrategy Inc.

The authentication user DN is specified in the Intelligence Server Configuration Editor, in the **LDAP**: **Server** category, in the **Distinguished Name (DN)** field under Authentication User.

The user's DN is specified on the User Editor in the **Authentication: Metadata** category in the **Distinguished Name (DN)** box under LDAP Authentication.

If no explicit link is specified, the LDAP user is imported as a new MicroStrategy user under the `LDAP Users` group if the **Import Users** check box is selected. The user can then be treated as any MicroStrategy user and assigned privileges. The user object in the metadata for the MicroStrategy user now also contains a link to the LDAP user after the import.

Intelligence Server also allows LDAP groups to be imported. With this option selected, all the groups to which the user belongs are also imported under the `LDAP Users` group (similar to the imported user) when an LDAP user logs in.

You cannot link MicroStrategy system groups (such as the **Everyone** group) to an LDAP group.

The hierarchical visual relationship between users and their user group is not maintained in the `LDAP Users` folder because it is maintained in the LDAP server directory. In spite of the visual link not appearing, the actual link between the user and his/her group does exist and is maintained.

The Synchronize at Login options for both users and groups cause Intelligence Server to check (at the time of next login) whether:

- MicroStrategy user information in the metadata and the LDAP user information in the LDAP Server are synchronous
- MicroStrategy group information in the metadata and LDAP group information in the LDAP Server are synchronous

The names and links between the two may or may not be synchronized depending on whether the synchronize option is selected in combination with whether users and groups are to be imported.

When the user is logged in as a temporary LDAP user/group:

- There is no link persisted in the metadata and the user has the privileges of a Guest user as long as the user is logged in. The user has the privileges of the MicroStrategy Public/Guest group.
- The non-imported user does not have an Inbox, because the user is not physically present in the metadata.
- The non-imported user cannot create objects and cannot schedule reports.

 © 2016, MicroStrategy Inc.

LDAP frequently asked questions

Do LDAP users have their own Inbox and Personal folders?

If users are imported into the metadata, they have their own Inbox and personal folders. If users are not imported, regardless of whether they are part of the LDAP Users or LDAP Public group, they do not have an Inbox. Users that are not imported do not have personal folders and can save items only in public folders if they have the correct privileges and permissions.

How can I assign security filters, security roles (privileges), or access control (permissions) to individual LDAP users?

Security filters, security roles, and access control may be assigned to users after they are imported into the MicroStrategy metadata, but this information may not be assigned dynamically from information in the LDAP repository.

To allow users to dynamically inherit this information, you should assign these permissions at the group level in the MicroStrategy metadata. Group membership information is dynamically determined each time an LDAP user logs into the system, according to the group they become part of.

May two different users have different LDAP links, but the same user name?

Yes. The MicroStrategy metadata may contain two users with the same user name as long as the login name is different. For example, one employee can use jsmith as his login name, and the other can use johnsmith as his login name. With these unique login names, each user can use John Smith as his user name.

What happens if there are two users with similar descriptions in the LDAP directory?

If the DN descriptor that specifies a user is not sufficient for Intelligence Server to identify him/her, the user will fail to log in. You should enhance the User search filter and Group search filter in the LDAP configuration category to help identify the user.

What happens if I import a User Group along with all its members in the LDAP directory into MicroStrategy metadata and then assign a connection mapping to the imported group?

The connection mapping of the imported user group to which the user belongs does not readily apply to the user. For this to work, you must manually assign the user as a member of the group after she has been imported.

© 2016, MicroStrategy Inc.

Fixing inconsistencies in the metadata

Logical inconsistencies in the metadata may arise because of a mismatch or errors in the definitions of the DSS objects stored in a metadata repository.

Fixing deleted items

If you attempt to use an object that has been deleted by another user during your session, you may receive an error message similar to the following:

```
'Object with ID '46E2C20D46100C9AFD5174BF58EB8D12' and
type 26(Column) is not found in the metadata. It may have
been deleted.'
```

You can verify that the object no longer exists in the project by disconnecting and reconnecting to the project.

The following table lists all the object types and object descriptions that occur in the MicroStrategy metadata. You can refer to the type of the missing object from the table and restrict your search to only that object. This way you do not have to search through all the objects in a project.

Object Type	Object Classification	Object Description
-1	DssTypeUnknown	The type of object is not specified
0	DssTypeReserved	None
1	DssTypeFilter	Filter
2	DssTypeTemplate	Template
3	DssTypeReportDefinition	Report
4	DssTypeMetric	Metric
5	Unused	None
6	DssTypeAutostyles	Autostyle
7	DssTypeAggMetric	Base formula
8	DssTypeFolder	Folder
9	Unused	None
10	DssTypePrompt	Prompt
11	DssTypeFunction	Function
12	DssTypeAttribute	Attribute

© 2016, MicroStrategy Inc.

Object Type	Object Classification	Object Description
13	DssTypeFact	Fact
14	DssTypeDimension	Hierarchy
15	DssTypeTable	Logical table
16	Unused	None
17	DssTypeFactGroup	Fact group
18	DssTypeShortcut	Shortcut (a reference to another object)
19	DssTypeResolution	Saved prompt answer
20	Unused	None
21	DssTypeAttributeForm	Attribute form
22	DssTypeSchema	Schema (the collection of objects that define the data warehouse structure)
23	DssTypeFindObject	Search definition (a simple search) **Note**: This object type is deprecated. Search objects are object type 39 (DssTypeSearch).
24	DssTypeCatalog	Catalog (a list of relevant tables in a database)
25	DssTypeCatalogDefn	Catalog definition (a description of how a catalog is constructed)
26	DssTypeColumn	Column (a property needed to define a column of a database table)
27	DssTypePropertyGroup	Property group (an internal object used to cache lists of property sets)
28	DssTypePropertySet	Property set (an internal object)
29	DssTypeDBRole	Database Instance
30	DssTypeDBLogin	Database Login
31	DssTypeDBConnection	Database Connection
32	DssTypeProject	Project
33	DssTypeServerDef	Server definition (a description of a configuration of an Intelligence Server)
34	DssTypeUser	MicroStrategy User or Group
35	Unused	None
36	DssTypeConfiguration	Intelligence Server configuration (a top level object representing a MicroStrategy installation)
37	DssTypeRequest	Scheduled request

© 2016, MicroStrategy Inc.

Object Type	Object Classification	Object Description
38	Unused	None
39	DssTypeSearch	Search object
40	DssTypeSearchFolder	Search folder (a folder-like object used to store the result of a search object)
41	Unused	None
42	DssTypeFunctionPackageDefinition	Function Definition
43	DssTypeRole	Transformation
44	DssTypeSecurityRole	Security Role
45	DssTypeInBox	History folder, or inbox (a folder-like object used to store History messages)
46	DssTypeInBoxMsg	History message (an object that describes the status of a report execution request)
47	DssTypeConsolidation	Consolidation
48	DssTypeConsolidationElement	Consolidation element
49	DssTypeScheduleEvent	Schedule event (an event that can trigger a scheduled object)
50	DssTypeScheduleObject	Scheduled object (an object that can be triggered)
51	DssTypeScheduleTrigger	Scheduled trigger (the binding between schedule event and object)
52	DssTypeLink	Link (the holder of a property that spans objects)
53	DssTypeDBTable	Physical database table
54	DssTypeTableSource	Same suffix and prefix tables (a collection of physical database tables with the same suffix and prefix)
55	DssTypeDocumentDefinition	Report Services Document
56	DssTypeDrillMap	Drill map
57	DssTypeDBMS	DBMS definition (an object that holds information about a physical database)
58	DssTypeMDSecurityFilter	Security filter

Manual editing of values in the MicroStrategy metadata incorrectly may cause serious, project-wide problems that may make the project unusable. Because these are user-initiated changes, they are not covered by any MicroStrategy warranty. Users are strongly encouraged to back up their metadata before any alteration.

© 2016, MicroStrategy Inc.

Scan MD

Scan MD is a utility that detects and repairs logical inconsistencies in the MicroStrategy metadata without requiring code-level changes to the core platform. The Scan MD tool searches for logical errors and fixes them within the platform object structure. The Scan MD tool executes tests to detect inconsistencies in the metadata. These tests are based on modules that are activated by keys. Each module can contain a number of tests and fixes to any problems that the test finds. The product is shipped with a default module-key pair that can run a standard set of tests to detect inconsistencies in your project. This default key is displayed in the Process options page of the wizard automatically. In some cases, Technical Support will provide you with a key-module pair to fix other issues that are not included in the standard tests. These keys and modules are provided to address a customer-specific issue and are not required for every customer. In such cases, it is always best to work with a support representative to understand the nature of your issue. The support representative will determine if there is a Scan MD module that addresses your problem.

Object dependencies troubleshooting

If you attempt to delete multiple objects at once in Developer or Object Manager, and the objects have parent/child dependencies on each other, you may get an error message. For example, you have an Intelligent Cube named “Time, products, geography” and a single view report that uses that Intelligent Cube, named “Yearly sales of electronics.” If you attempt to delete both these objects at the same time, an error message pops up saying that you cannot delete the Intelligent Cube because other objects depend on it.

This occurs because MicroStrategy products handle multiple object deletions in alphabetical order. To resolve this, delete the view report first, and then delete the Intelligent Cube.

Date/time functions troubleshooting

In some cases, MicroStrategy no longer updates certain DBMS objects as newer versions are released. Although MicroStrategy does not normally remove these, in one case, we merged “Oracle 8i R2” and “Oracle 8i R3” into “Oracle 8i R2/R3” for both Standard and Enterprise editions (Oracle 8i R3 is no longer being updated). You may need to select the merged version as part of your database instance if you are using a version of Oracle 8i. This will become apparent if date/time functions stop working, particularly in Enterprise Manager. For details, see *Upgrading your database type properties, page 11* in *Introduction to MicroStrategy System Administration*.

© 2016, MicroStrategy Inc.

Performance troubleshooting

Project performance

You may notice a project that takes longer to load than others, or you may see the following error message in the server log file:

```
[DSS Engine] [Error]DSSSQLEngine: WARNING: Object cache
MaxCacheCount setting (200) is too small relative to
estimated schema size (461).
```

The project loading time involves retrieving a significant amount of data and can be time-consuming depending on the project's size, your hardware configuration, and your network. If a project takes a long time to load or you see the error message above, you can look at the following:

- **Load at startup**: During installation of MicroStrategy, did you select to have the project "Load at startup"? This option generally speeds subsequent requests for the project because much of the information necessary to load the project is cached on the server.

  You can check whether your project is set to load at startup using the Intelligence Server Configuration Editor. (In Developer, right-click the project source and select **Configure MicroStrategy Intelligence Server**. Expand the **Projects** category, and then select **General**.)

- **Maximum RAM usage for object cache**: Depending on the size of your schema objects, you may need to raise the Maximum RAM usage (KBytes) setting. (To locate this setting: In Developer, right-click on the project and select **Project Configuration**. From the **Caching** category, select **Objects**.)

  The error message shown above will contain an estimated schema size. This number should be multiplied by 10 and the result entered in the Maximum RAM usage setting. This setting may help optimize your project.

  Raising the number for Maximum RAM usage may cause high memory use, which may cause your machine to experience problems. Be sure you understand the full ramifications of all settings before you make significant changes to them.

- **Turn off diagnostic tracing**: If you have tracing turned on, turn it off to ensure the project is loaded without logging any unnecessary information that can slow down performance. (In Developer, from the **Tools** menu, select **Diagnostics**. Click **Help** for details on the various tracing options.)

 © 2016, MicroStrategy Inc.

Schema update performance

If the schema update is taking a long time, you may need to increase the amount of memory available to the object cache on Intelligence Server. When the schema for a project is updated, Intelligence Server accesses all schema objects. If the object cache does not have enough memory to hold all the schema objects, some must be swapped in and out of the object cache, slowing down the update process. For more information about object caches, see *Object caches, page 451*.

To increase the amount of memory available to the object cache, in the Project Configuration Editor, expand the **Caching: Auxiliary Caches: Objects** category, and increase the value in the **Server**: **Maximum RAM usage (MBytes)** field.

You can use the Diagnostics and Performance Logging Tool to trace the schema update process and determine whether you need to increase the memory available to the object cache. In the Diagnostics and Performance Logging Tool, enable the Metadata Server > Transaction and Engine > Scope traces. For information about the Diagnostics and Performance Logging Tool, see *Finding trouble spots using diagnostics, page 731*.

Cache Monitor and Intelligent Cube Monitor performance

If you are running Intelligence Server on HP-UX v2 (B.11.23), you may notice a slow response time when using the Cache Monitor or the Intelligent Cube Monitor. This is due to an issue with the design of the HP-UX v2 file system, and has been fixed in HP-UX v3 (B.11.31).

You can improve the response time of the Cache Monitor and the Intelligent Cube Monitor on HP-UX v2 by changing the following settings on the Intelligence Server machine (requires root access):

```
# kctune nfs_new_lock_code=1
# kctune nfs_async_read_avoidance_enabled=1
# kctune nfs_fine_grain_fs_lock=1
# kctune nfs_new_rnode_lock_code=1
# kctune nfs_wakeup_one=1
```

Troubleshooting data imported from a file

You can use MicroStrategy Web to import data from a file, such as a text file, CSV file, or Excel spreadsheet. The imported data is stored in an Intelligent Cube. For more information about importing data, including instructions, see the *Project Design Guide* or the MicroStrategy Web Help.

- When importing a large file, make sure that sufficient memory is available on both the client and server machines:
    - On the client machine, that is, the machine from which the user initiates the import, the web browser may require an amount of memory up to five times the size of the file being imported. For example, if you are importing a 50 MB Excel file, the web browser may use up to 250 MB of RAM.
    - On the Intelligence Server machine, Intelligence Server may require an amount of memory up to four times the size of the file being imported. For example, if you are importing a 50 MB Excel file, Intelligence Server may use up to 200 MB of RAM.
- On the Intelligence Server machine, ensure that the Java Virtual Machine's memory limit is set to 800 MB. For instructions, see the documentation provided for the Java Virtual Machine.
- You may need to increase the timeout settings for your network, web browser, or web server.

Subscription and report results troubleshooting

Drilled-to report returns no data or incorrect data

If you are using In-memory services, you may drill to a report and find that the new report does not return any data, or returns subtotals that are inconsistent with the subtotals on the parent report. This is because the parent report is using an Intelligent Cube for its data source, and that Intelligent Cube does not contain data at the level you have drilled to.

You can confirm whether a report is connecting to an Intelligent Cube in SQL View. If the SQL View contains the line `Intelligent Cubes Accessed:` the report is connecting to one or more Intelligent Cubes. The Intelligent Cubes the report is connecting to are listed below that line.

You can disable the use of Intelligent Cubes for a report in the report's VLDB properties. In the VLDB Properties dialog box, in the **Advanced** category, under **Dynamic Sourcing**, clear the **Enable Dynamic Sourcing** check box. Be aware that this causes the report to use the data warehouse instead of the Intelligent Cube, which may cause an unexpected load on the data warehouse.

 © 2016, MicroStrategy Inc.

Graph report results

Depending on the graph style users choose to display their data, graph reports may or may not show all the features expected from the chosen graph style. Each graph style has its own requirements for type of data displayed, individual options set, and combinations of settings enabled.

For details on many of the graph styles available in MicroStrategy and the specific requirements for each style, see the *Graphing* appendix in the *Advanced Reporting Guide*.

Number of report result rows

You can configure MicroStrategy at the project level to limit the number of report rows to be returned from the data warehouse and displayed to the user. This setting is configurable in the Project Configuration Editor. (In Developer, right-click a project, select **Project Configuration**, expand **Governing Rules**, expand **Default**, and select **Result sets**. The setting is Number of report result rows.)

If you change this setting, reports may still continue to be returned and displayed showing more than the maximum number of report result rows allowed. This is because this setting is designed to apply only to reports created after this setting has been changed. When you change this setting, no existing reports are affected.

You can limit the number of report result rows on existing (and new) reports individually, using the **Result Set Row Limit** VLDB property. The Result Set Row Limit VLDB property can be specified for any report. For instructions on setting VLDB properties, see the *VLDB Properties* chapter in the *Supplemental Reference for System Administration*.

Subscription with a prompt is not sent and disappears

If you use a prompt on a report, and that prompt requires an answer but no default answer is provided in the prompt's design, and then that report is subscribed to, the subscription cannot execute the report successfully because the prompt requires an answer. The subscription is therefore invalidated and automatically removed from Developer's Subscriptions list.

For the subscription to work, you can redesign the prompt so that it has a default answer, or simply remove the prompt from the report.

Contact does not appear in list of contacts when subscribing

- A contact must have at least one address for the subscription type that you want to create. For example, if you want to subscribe a contact to receive a report via email, that contact must have at least one email address associated with the contact, otherwise the contact does not appear on the subscription's list of contacts. If you

© 2016, MicroStrategy Inc.

want to subscribe the contact to receive a delivery via printer, that contact must have a printer path (address). Use the Contact Editor to add addresses (see *Creating and managing contacts, page 522*).

- A contact must be linked to a MicroStrategy user for that contact to appear in any list of contacts. Use the Contact Editor to link the contact to a user (see *Creating and managing contacts, page 522*).
- A subscription or schedule owner must have the Use permission and the Browse permission for a contact to see that specific contact in any list of contacts.

SQL/MDX string length

When you generate a report using Developer, you might see the following error message:

```
Error: SQL Generation Complete[]QueryEngine encountered
error: The generated SQL/MDX string (xxxxx bytes) is
longer than the limitation of the corresponding connector
(yyyyy bytes).[]Error in Process method of Component:
QueryEngineServer, Project ###, Job ###, Error Code=-
2147212544
```

This error message is displayed when the SQL string size exceeds the maximum value set for the SQL/MDX string.

You can increase the maximum value for the SQL/MDX field in the Project Configuration dialog box as follows:

1. In Developer, right-click the project and select **Project Configuration**.
2. In the Project Configuration Editor, expand the **Database instances** category and select **SQL Data Warehouses**.
3. Click **VLDB Properties**. The VLDB Properties dialog box opens.
4. Expand **Governing** and select the **Maximum SQL/MDX Size** VLDB setting.
5. Clear the **Use default inherited value - (Default Settings)** check box.
6. Increase the **Maximum SQL/MDX Size** value as required. You can enter any number between 1 and 999999999. If you enter 0 or -1, the **Maximum SQL/MDX Size** is set to the default value of 65536. This default size may be different for different databases. It depends on the database instance that you select.

You should enter a value that a certified ODBC driver can handle; a large value can cause the report to fail in the ODBC driver. This is dependent on the database type you are using.

If increasing the value of this VLDB property does not resolve the issue, try simplifying the report. You can simplify a report by removing attributes, metrics, and filters. Importing large sets of elements for filters can often cause large SQL/MDX size.

 © 2016, MicroStrategy Inc.

For more information, see the *VLDB Properties* chapter in the *Supplemental Reference for System Administration*.

Freeform SQL report error

When you run a Freeform SQL report, you might see the following error message:

```
Error: SQL Generation CompleteQueryEngine encountered
error. Execute Query failed. Error type: ODBC error. ODBC
operation attempted: SQLExecDirect. [37000-3100 on HSTMT]
[Microsoft][ODBC Excel driver] Syntax error [missing
operator] in query expression `pa0.DAY_DATE WJXBFSO`
Error in process method of Component: QueryEngineServer,
Project MicroStrategy Tutorial, Job 1098, Error Code=-
2147212544.
```

This error message can result from an incorrect setting in the database instance. If the database instance is using a Microsoft Excel file as a data source and the database instance type is set to Generic DBMS, there is a change in the syntax. This change in the syntax generates the error message.

To avoid this, change the **Database connection type** of the database instance to Microsoft Excel 2000/2003 as follows:

1. From the Developer Folder List, expand **Administration**, and then select **Database Instance Manager**.
2. Right-click the name of the database instance that you want to modify and select **Edit**. The Database Instance dialog box opens.
3. On the **General** tab, select the **Database connection type** as Microsoft Excel 2000/2003.
4. Click **OK**.

For the change to take effect, you must restart the Intelligence Server that uses this database instance.

Internationalization troubleshooting

For detailed information about internationalization, including instructions on creating, enabling, and disabling languages for objects and projects, see the *Internationalization* chapter in the *Supplemental Reference for System Administration*.

Object displays translations in a language different from the project or object default language

If the object does not contain translations for either the project default language or the object default language, it must use the translations that it does have. The system treats those object translations as if they were in the project default language.

For example, the project's default language is English, the object's default language is French, and the object contains only French translations. If French is disabled from the project, the French translation is displayed but is be treated by the system as if it were English. The object's default language automatically changes to English.

User's preferred language defaults to project default language

When a language is disabled for a project or removed from a project, and a user has that language selected as his preferred language, the user sees the project default language. The user can change his preference to any available language at any time.

Custom names for system objects are not retained after upgrade

After upgrading a project's system object translations in the Project Configuration Editor, any system objects whose names have been customized revert to their predefined MicroStrategy names.

To retain your custom names for system objects after a project upgrade, use Configuration Wizard to upgrade the project instead of upgrading it through the Project Configuration Editor. For instructions on using Configuration Wizard to upgrade a project, see the *MicroStrategy Upgrade Guide*.

Troubleshooting Intelligence Server

Logon failure

When you attempt to start Intelligence Server, you may receive the following error message:

```
Failed to start service
```


© 2016, MicroStrategy Inc.

```
Unexpected error occurred (::StartService() failed

System Error 0x0000042D (1069): The service did not
start due to a logon failure.)
```

This issue can occur because the password for the account used to automatically start Intelligence Server has changed. MicroStrategy Service Manager does not automatically update the stored password, so you must manually update this password.

To change the stored password for Intelligence Server

1. Open the MicroStrategy Service Manager.
   - In Windows: double-click the Service Manager icon in the system tray. If the Service Manager icon is not present, from the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Tools**, and then select **Service Manager**.
   - In UNIX: You must be in an XWindows environment to run Service Manager in UNIX. From the `/bin` directory in the MicroStrategy directory, type `./mstrsvcmgr` and press **ENTER**.
2. From the **Service** drop-down list, select **MicroStrategy Intelligence Server**.
3. Click **Options**. The Server Options dialog box opens.
4. In the **Password** and **Confirm Password** fields, enter the correct password.
5. Click **OK**. The Server Options dialog box closes.
6. To start Intelligence Server, click **Start**.

Failure to activate Intelligence Server

When you try to start Intelligence Server using the MicroStrategy Service Manager, you might get the following error message:

```
Failed to start service

Error code: -1

Error Message: Intelligence Server cannot start because
it has not been activated with MicroStrategy. Please
activate your installation via License Manager or by
visiting the activation web site at
https://licensing.microstrategy.com.
```

This error occurs when Intelligence Server is not activated, either because it has not been activated in the activation grace period or because it has become deactivated.

If Intelligence Server has not been activated, you can start it by activating the license for Intelligence Server installation. For more information on activating the Intelligence Server installation, see the *MicroStrategy Installation and Configuration Guide*.

© 2016, MicroStrategy Inc.

If Intelligence Server has previously been activated, it may have become deactivated because the unique machine identifier information has changed or is no longer accessible. On a Windows machine, the unique machine identifier is the Network Ethernet Card MAC address. So, for example, the machine may have multiple network cards, and the primary network card may be disabled. In this case, the MAC address that identifies the machine is the address of a different network card, and so Intelligence Server becomes deactivated.

A related error message when attempting to start Intelligence Server is:

```
2147205286: Fail to authenticate the licensed feature
```

In this case, the system date on the Intelligence Server machine is incorrect and is beyond the expiration date of the installation key. To start Intelligence Server, correct the date on the Intelligence Server machine and then restart the machine.

Multiple instances of Intelligence Server

You may see multiple instances of Intelligence Server running on the same machine.

This can occur when the hard drive on the Intelligence Server machine is completely full. To prevent this problem, ensure that the Intelligence Server machine always has at least 1 MB of free space on the hard drive.

Clustered environments troubleshooting

Problems in a clustered environment

All nodes must join the cluster before you make any changes to any governing settings, such as in the Intelligence Server Configuration Editor.

What order should I start the server nodes?

If powering on after a power failure, it does not matter what order the nodes are powered back on. Once the nodes are on, use the Cluster Monitor to determine which machine is the primary node, and then manage any caches.

 © 2016, MicroStrategy Inc.

Which node am I connected to?

In Developer, the project source definition includes the server name. Developer connects to a specific node only, so you can control which server you are connected to.

In MicroStrategy Web, new connections are dynamically directed to the least loaded node. Although a server name may appear as part of the URL, this may not be the node to which the user is connected. To determine which node a Web user is connected to, use the User Connection Monitor to see which node the user's name appears in (see *Monitoring users' connections to projects, page 57*).

Do I need multiple project sources?

Because Developer connects to a specific node in a cluster, project sources must be created to each node to fully access and monitor the cluster.

Does it matter which node I connect to in Developer?

Nodes in the cluster can host a different set of projects from the same metadata. The node to which a Developer user connects is important because it dictates which projects are available to the user at that time.

Can I control which node I connect to in MicroStrategy Web?

Controlling connections other than by default load balancing requires customizing MicroStrategy Web. The MicroStrategy SDK and MicroStrategy Developer Library (MSDL) contain information on customizing MicroStrategy Web.

Caches in a clustered environment

Did my report hit a cache?

If the cache is available on the local node, the Cache Monitor increments the hit count. If the cache is retrieved from another node, speed of response can indicate whether a cache is hit. Statistics tables can provide additional data on cache hits.

What cache did my report hit?

The statistics logs show cache hit statistics, logged to the STG_IS_CACHE_HIT_STATS table. This information is later loaded in the IS_CACHE_HIT_STATS table as part of the Enterprise Manager data load process.

© 2016, MicroStrategy Inc.

How do I automate the deletion of caches on multiple nodes?

Use MicroStrategy Command Manager to run scripts that automatically delete caches. See *Chapter 15, Automating Administrative Tasks with Command Manager*.

Node synchronization in a clustered environment

When do I need to purge the object cache?

When an object is edited on one cluster node, the updated version ID of the object is announced to the other nodes. This allows the other nodes to invalidate the object if they have it in memory and retrieve a fresh copy from the metadata. Therefore, in this instance there is no need to purge the object cache.

If changes to an object are made in 2-tier (Direct) mode, those changes are not propagated to any Intelligence Servers connected to the metadata. Additionally, if an object is modified from an Intelligence Server that is not in the cluster but it is using the same metadata, the cluster nodes will not know of the object change. In these cases, the object cache should be purged.

When do I need to purge the element cache?

The element cache should be purged as a routine part of a warehouse load or any time the elements associated with an attribute may have changed.

You can automate purging the element cache using MicroStrategy Command Manager. See *Chapter 15, Automating Administrative Tasks with Command Manager*.

Which machine is my History List on?

For Intelligence Server 7.1 and later, the combined History List in memory is a sum of all local files and is automatically synchronized. Therefore, you cannot tell which pointers are on which machine.

In what order should I start the server nodes?

If powering on after a power failure, it does not matter what order the machines are started. It is important to locate the machine that is the primary node, so that cache management can be controlled. The primary node is designated in the Cluster Monitor. See *Managing your clustered system, page 390*.

 © 2016, MicroStrategy Inc.

Statistics logging troubleshooting

Each project in a MicroStrategy system can be configured to log usage statistics to a database. For detailed information about statistics logging, see *Monitoring system usage: Intelligence Server statistics, page 288*.

Statistics are not being logged

If statistics are being logged but some data is being lost, the load on your system may be too high. For ways to decrease the system load, see *Chapter 8, Tuning Your System for Best Performance*.

If you are logging statistics to a DB2 database, disabling the "Application Using Threads" setting for the statistics DSN may improve performance on AIX systems. For more information and detailed instructions, see your database documentation.

For troubleshooting tips in cases where statistics suddenly stop being logged, see *Statistics logging suddenly stops, page 779*.

If statistics do not appear to be logged for your project, first verify that the Intelligence Servers to be monitored are correctly logging information in the statistics tables and that these tables are in the correct locations in the statistics and Enterprise Manager repository. The table names beginning with `STG_` are the staging tables. The Intelligence Server logs information there until the Enterprise Manager data load process moves the information to the table of the same name without the `STG_` prefix.

When the statistics tables are created using the MicroStrategy Configuration Wizard, they must be created in the database to be used as the Enterprise Manager warehouse.

The following Structured Query Language (SQL) scripts can confirm whether statistics are being recorded correctly. They should be executed using a native layer query tool (for example, SQL+ for Oracle, Query Analyzer for SQL Server).

- Check statistics logged by user sessions:
```
SELECT * FROM STG_IS_SESSION_STATS;
SELECT * FROM IS_SESSION_STATS;
```
- Check report-related statistics:
```
SELECT * FROM STG_IS_REPORT_STATS;
SELECT * FROM STG_IS_REP_STEP_STATS;
SELECT * FROM STG_IS_REP_SQL_STATS;
SELECT * FROM STG_IS_REP_SEC_STATS;
SELECT * FROM IS_REPORT_STATS;
SELECT * FROM IS_REP_STEP_STATS;
SELECT * FROM IS_REP_SQL_STATS;
SELECT * FROM IS_REP_SEC_STATS;
```
- Check schedule-related statistics:
```
SELECT * FROM STG_IS_SCHEDULE_STATS;
SELECT * FROM IS_SCHEDULE_STATS;
```

© 2016, MicroStrategy Inc.

- Check cache-related statistics:
```
SELECT * FROM STG_IS_CACHE_HIT_STATS;
SELECT * FROM IS_CACHE_HIT_STATS;
```

- Check document-related statistics:
```
SELECT * FROM STG_IS_DOCUMENT_STATS;
SELECT * FROM STG_IS_DOC_STEP_STATS;
SELECT * FROM IS_DOCUMENT_STATS;
SELECT * FROM IS_DOC_STEP_STATS;
```

For a detailed list of the statistics tables and columns, see *Enterprise Manager Data Model and Object Definitions* in the *Supplemental Reference for System Administration*.

If data is not being logged to statistics, or the information is not up to date, possible reasons include:

- Intelligence Server has shut down while jobs were still executing. Statistics are not logged for jobs that do not complete before Intelligence Server shuts down. This applies to both a manual shutdown and to an unexpected shutdown.
- The statistics database is not selected in the Project Configuration dialog box.

  To check this, right-click a project that should be logging statistics and select **Project Configuration**. Then select the **Statistics** category and confirm that the correct database instance is selected.
- The statistics database instance has not been configured correctly.

  To check this, expand the **Administration** section for the project source that should be logging statistics. Select **Database Instance Manager**. Right-click the statistics database instance and select **Edit**. Then verify the database connection type, data source name (DSN), and default database connection.
- The statistics database DSN has not been configured correctly.

  To check this, open the Windows Open Database Connectivity (ODBC) Data Sources for the machines where the Intelligence Servers logging statistics are installed. Verify that the DSN used by the Statistics Database Instance are defined correctly using certified drivers to connect to the statistics and Enterprise Manager repository. For a list of certified databases, see the *MicroStrategy Readme*.
- If the Intelligence Servers for the projects being monitored and the database server do not have synchronized clocks, some data may be incorrect in the statistics tables. For example, if statistics appear for “Deleted report” in Enterprise Manager reports, it may be because statistics are being logged for reports that according to the warehouse’s timestamp should not exist. For additional information about the need to synchronize the data warehouse and the Intelligence Servers, see the *MicroStrategy Operations Manager Guide*.
- If you are using single instance session logging and the specified project is not one of the monitored projects, no data for any monitored project is logged. To check this, open the Intelligence Server Configuration Editor to the **Statistics: General** category and confirm that the project selected in the single instance session logging drop-down list is one of the monitored projects. For more information about single

 © 2016, MicroStrategy Inc.

instance session logging, see *Logging all statistics from a project source to the same database, page 291*.

Statistics logging suddenly stops

If statistics logging for a project suddenly stops, one or more of the following factors may be the cause:

- The database server hosting the statistics database may not be running.
- The database connection from Intelligence Server to the statistics database server may have failed.
- A heavy load on the statistics database may have caused statistics logging to shut down.
- The login or password for the statistics database may have been modified or expired.
- Intelligence Server may have stopped logging statistics. For possible actions you can take to correct this, see *Statistics are not being logged, page 777*.

Statistics on subscriptions and deliveries

MicroStrategy's Narrowcast Server product logs statistics about each of the Narrowcast services and subscriptions, under the Transmission Recording component. Similar information is logged as part of the statistics collected for MicroStrategy Distribution Services.

The information is logged in the schedule statistics staging table, STG_IS_MESSAGE_STATS. The Enterprise Manager data load process moves the information to the IS_MESSAGE_STATS table. The details of the information in this staging table are in the *Statistics Data Dictionary* in the *Supplemental Reference for System Administration*. The table below shows the pertinent Narrowcast Server information and the equivalent information in the Enterprise Manager statistics tables for Distribution Services.

Narrowcast Server Service Information Category	Distribution Services Schedule column names in STG_IS_MESSAGE_STATS and IS_MESSAGE_STATS tables
Email	PHYSICALADDRESS
User ID	RECIPIENTCONTACTID or RECIPIENTCONTACTNAME
Login ID	Not applicable. All users log in as themselves and use USERID.
Unique Message ID	Create a unique message ID by combining MESSAGEINDEX with SESSIONID
Subscription Set ID	SCHEDULEID. The Narrowcast Server concept of a subscription set is a schedule in Distribution Services.
Address ID	ADDRESSID
Subscription ID	SUBSINSTID

Additional resources for troubleshooting

You can improve your troubleshooting efforts by using all the troubleshooting best practices and information resources available. Information to access each resource is described below.

MicroStrategy Readme and release notes

The *MicroStrategy Readme* and release notes provide, among other things, thorough coverage of system requirements for every MicroStrategy product, detailed lists of issues that have been fixed by each release, and links to additional resources for further information on a topic. They are well-organized, indexed, and provide a left-hand menu listing related information for whatever you are currently viewing. The Readme and release notes are often overlooked as a primary source of system configuration and troubleshooting information.

At the end of your MicroStrategy installation, you were prompted to view the *MicroStrategy Readme*, which also includes release notes. You can access the Readme in the following ways:

- **Windows**: From the Windows **Start** menu, point to **All Programs**, then **MicroStrategy Documentation**, and then select **ReadMe**.
- **UNIX/Linux**: Double-click the **readme.htm** file in the `install` folder where you installed MicroStrategy.

Readme

The *MicroStrategy Readme* provides general information on the MicroStrategy release and information on the MicroStrategy products, pertaining to:

- **System requirements**: MicroStrategy products require certain hardware, software, and other system requirements to function correctly. If your system does not have the necessary requirements for a MicroStrategy product, the component may encounter errors, loss of functionality, poor performance, and so on.
- **Compatibility and interoperability**
- **Certified and supported configurations**

Release notes

The release notes supplied for each MicroStrategy product provide:

- **Known issues**: You can review any functionality that has been identified as having a known issue for the related version. A description of the issue is given, and in many cases work-arounds are provided.

 © 2016, MicroStrategy Inc.

- **Resolved issues**: You can review any issues that have been resolved for the related version.
- **Troubleshooting**: You can review a list of troubleshooting tips for each MicroStrategy component. The troubleshooting tips may include a description of the symptom of the issue, the cause of the issue, and a resolution for the problem.

MicroStrategy Knowledge Base

The Knowledge Base is a repository of technical documentation, designed to provide a level of customer support 24 hours, 7 days a week. The Knowledge Base consists of information such as:

- Specific troubleshooting documents
- Release notes
- All user manuals
- White papers
- Newsletters

The troubleshooting documents that are included in the Knowledge base are created by MicroStrategy developers, engineers, and consultants. You can find helpful tips and solutions for various MicroStrategy error codes and issues.

The Knowledge Base is available to all MicroStrategy employees, distributors, partners, VARS, and customers with licensed, active maintenance agreements. It is also available to all evaluating customers for the duration of the evaluation period. You can access the Knowledge Base by navigating to the MicroStrategy Support page at this URL: https://resource.microstrategy.com/Support/.

Searching the Knowledge Base

The single field at the top of the support site is where you type a search word or phrase to search the entire Knowledge Base. You can do a simple search for keywords or phrases or choose an **Advanced Search** at the top right and include search criteria such as known issues, document status, last modification date, and so on.

Use these best practices to improve the results of your Knowledge Base search:

- If you are troubleshooting an error message or error code, you can copy the error message into your Knowledge Base search. You may have to enclose the error message in double quotes (“ ”) if it includes certain characters.
- If a specific search for information does not return helpful results, simplify your search terms and use an Advanced Search to select specific groups of information sources. This searches more generally but within a specific set of information.
- You can choose the software version in the advanced search options, but you can also include the version in your search terms. For example, you can type product line keywords such as 9.x, 9.2.1, and so on.

Customer Forums

The MicroStrategy Customer Forums is a group of message boards where MicroStrategy customers can have open discussions on implementation experiences, troubleshooting steps, and any fixes or best practices for MicroStrategy products. You can post and respond to message threads. The threads can help answer questions by pooling together one or more experiences and solutions to an issue.

The Customer Forums are not meant to replace Technical Support, and, while MicroStrategy employees monitor the forums from time to time, they are not responsible for answering messages posted to the Customer Forums.

You can access the Customer Forums from the left side menu of the Knowledge Base site, or from this URL: https://forums.microstrategy.com.

You are prompted to create a Customer Forums account if you have not already done so. For those customers with an active MicroStrategy account who are also designated as Support Liaisons for their organizations, you may already have access to post and view messages on the Customer Forums.

Searching the Customer Forums

The Customer Forums cover many areas of MicroStrategy software, as described below:

- **Products**: These forums are grouped by MicroStrategy products such as Developer, Intelligence Server, Web, and so on.
- **Database**: Discussion thread topics cover different databases that can be used with MicroStrategy such as Oracle, DB2, Informix, and so on.
- **Customization (SDK)**: Discussion thread topics cover customizations to ASP, Java/JSP, and XML/XSL.
- **Miscellaneous**: These forums cover data modeling, statistical analysis, portals, Web Services, and other general topics.

The Customer Forums also provide a search field at the top right, which you can use to search for keywords, author, date, and so on.

Technical Support

MicroStrategy Technical Support helps to answer questions and troubleshoot issues related to your MicroStrategy products. If the troubleshooting resources described above do not provide you with a viable solution to your problem, you can call Technical Support to help troubleshoot your products.

 © 2016, MicroStrategy Inc.

GLOSSARY

access control list (ACL)

A list of users and groups and the access permissions that each has for an object.

active user

A user who executes a report or uses the system in some way.

See also **concurrent user**.

application object

A MicroStrategy object used to provide analysis of and insight into relevant data. Application objects are developed in Developer and they are the building blocks for reports and documents. Application objects include these object types: report, document, template, filter, metric, custom group, consolidation, prompt.

authentication

The system process of validating user login information. A login ID and password are compared against an authorized list, and if a match is detected, specific access rights and application privileges are granted to the user.

cache

A special data store holding recently accessed information for quick future access. This is normally done for frequently requested reports, which execute faster because they need not run against the data warehouse. Results from the data warehouse are stored separately and can be used by new job requests that require the same data. In the MicroStrategy environment, when a user runs a report for the first time, the job is submitted to the database for processing. However, if the results of that report are cached, the results can be returned immediately without having to wait for the database to process the job the next time the report is run.

© 2016, MicroStrategy Inc.

change journaling

The process of logging information about changes to objects in a project. The logged information includes items such as the user who made the change, the date and time of the change, and the type of change (such as saving, copying, or deleting an object).

cluster

A collection of two or more machines that provide services to a common set of users. Each machine in the cluster is called a node.

See also **clustering**.

clustering

A configuration strategy that provides uninterrupted access to data, enhanced scalability, and increased performance for users.

See also **cluster**.

concurrent user

A user who logs in to a MicroStrategy system. When a user logs in to the system, a user session is established and remains open until the user logs out of the system or the system logs the user out. Users that are logged in but are not doing anything still consume some resources on Intelligence Server.

See also **active user**.

configuration object

A MicroStrategy object appearing in the system layer and usable across multiple projects. Configuration objects include (among others) these object types: users, database instances, database login IDs, schedules.

connection borrowing

Occurs when Intelligence Server executes a job on a lower priority connection because no connections that correspond to the job's priority are available at execution time. High priority jobs can run on high, medium, and low priority connections. Medium priority jobs can run on medium and low priority connections.

connection mapping

The process of mapping MicroStrategy users to database connections and database logins. For MicroStrategy users to execute reports, they must be mapped to a database connection and database login.

connector

A step in a System Manager workflow that determines the logical order of the workflow based on the exit code of the process the connector is coming from.

cookie

A piece of information that is sent to your Web browser—along with an HTML page—when you access a Web site or page. This information is usually used to remember details

© 2016, MicroStrategy Inc.

about what a user did on a particular site or page for the purpose of providing a more personal experience for the user.

data source name (DSN)

Provides connectivity to a database through an ODBC driver. A DSN generally contains host machine name or IP address, instance name, database name, directory, database driver, User ID, password, and other information. The exact information included in the DSN varies by DBMS. Once you create a DSN for a particular database, you can use it in an application to call information from the database.

There are three types of DSNs:

- System DSN: can be used by anyone who has access to the machine. DSN information is stored in the registry.
- User DSN: is created for a specific user. Also stored in the registry.
- File DSN: DSN information is stored in a text file with a .DSN extension.

data warehouse

1. A database, typically very large, containing the historical data of an enterprise. Used for decision support or business intelligence, it organizes data and allows coordinated updates and loads.

2. A copy of transaction data specifically structured for query, reporting, and analysis.

database connection

Stores all data warehouse specific connection information such as DSN, driver mode and SQL execution mode as well as connection caching information.

database instance

1. A MicroStrategy object created in Developer that represents a connection to the warehouse. A database instance specifies warehouse connection information, such as the data warehouse DSN, Login ID and password, and other data warehouse specific information.

2. Database server software running on a particular machine. Though it is technically possible to have more than one instance running on a machine, there is usually only one instance per machine.

database login

Stores the login ID and password that Intelligence Server uses to connect to a particular database.

See also:

- login ID
- password

© 2016, MicroStrategy Inc.

database management system (DBMS)

A collection of programs that enables you to store, modify, and extract information from a database.

dataset

A MicroStrategy report that retrieves data from the data warehouse or cache. It is used to define the data available on a document.

diagnostics

The process of logging and analyzing the information on the operational performance of Intelligence Server.

distinguished name (DN)

The unique identifier of an entry in the LDAP directory.

document

1. A container for objects representing data coming from one or more reports, as well as positioning and formatting information. A document is used to format data from multiple reports in a single display of presentation quality.

2. The MicroStrategy object that supports the functionality defined in (1).

document instance

Facilitates the processing of a document through Intelligence Server like a report instance is used for reports. It contains the report instances for all the dataset reports and therefore has access to all that may be included in the dataset reports, including prompts, formats, and so on.

element browsing

The process of navigating through hierarchies of attribute elements. For example, viewing the list of months in a year.

element cache

Most-recently used lookup table elements that are stored in memory on Intelligence Server or Developer machines so they can be retrieved more quickly.

encryption

The translation of data into a sort of secret code for security purposes.

entry process

Any process in a workflow that can be selected as the first process to attempt in a System Manager workflow.

exit code

A code provided in a System Manager workflow provided when a process is completed. It provides information on whether the process was successful, and why.

© 2016, MicroStrategy Inc.

exit process

Any process that allows you to end a System Manager workflow. An exit process also explains how the workflow ended.

Extraction, Transformation, and Loading (ETL)

1) The process used to populate a data warehouse from disparate existing database systems.

2) Third-party software used to facilitate such a process.

failover support

Ensures that a business intelligence system remains available for use in the event of an application or hardware failure. Clustering provides failover support in two ways: load distribution and request recovery.

firewall

A type of technology that enforces an access control policy between two systems. It can be thought of as something that exists to block certain network traffic while permitting other network traffic.

group

(short name for "user group") A collection of users, such as Everyone, System Administrators, LDAP users, and so on. Groups provide a convenient way for managing a large number of users. You can assign privileges to groups as well as permissions to objects.

high watermark (HWM)

A value used by Intelligence Server to calculate the available memory for a new memory request for both virtual memory and virtual bytes (if it has exceeded an acceptable level). The HWM represents the highest value that the sum of private bytes and outstanding memory contracts may reach before triggering memory request idle mode. This defaults to 90%, but you can specify a lower value in the Maximum use of virtual address space (%) setting.

See also **low watermark (LWM).**

history cache

Report results saved for future reference via the History List by a specific user.

History List

A folder where users put report results for future references.

HTML document

1) A container for formatting, displaying, and distributing multiple grid and graph reports from a single request.

2) The MicroStrategy object that supports such functionality.

© 2016, MicroStrategy Inc.

HTML document instance

Facilitates the processing of the HTML document through Intelligence Server (like a report instance is used for processing reports). It contains the report instances for all the child reports, the XML results for the child reports, and any prompt information that may be included in the child reports.

idle time

The time during which a user stops actively using a session, for example, not using the project, not creating or executing reports.

inbox synchronization

The process of synchronizing inboxes across all nodes in the cluster so that all the nodes contain the same History List messages.

integrity test

A test performed in Integrity Manager. Reports from a base project are executed and you are informed as to which reports failed to execute. Depending on the type of integrity test, those reports may be compared against reports from another project, or against reports from a previously established baseline.

Intelligent Cube

A data structure containing data from the data warehouse that is stored in memory. Executing a report against an Intelligent Cube is faster and causes less database load than executing the report against the data warehouse. Intelligent Cubes are part of the In-memory Services add-on for Intelligence Server.

job

A request to the system, created by users submitting requests from Developer, Web, Narrowcast Server, Intelligence Server's internal scheduler, or a custom-coded application. Common requests include report execution, object browsing, element browsing, Report Services document execution, and HTML document execution. Job processing involves several procedures, depending on the specific request.

job priority

Defines the order in which jobs are processed.

Lightweight Directory Access Protocol (LDAP)

An open standard Internet protocol running over TCP/IP and designed to maintain and work with large directory services. An LDAP directory can be used to centrally manage users in a MicroStrategy environment by implementing LDAP authentication.

load balancing

A strategy aimed at achieving even distribution of MicroStrategy Web user sessions across Intelligence Servers. MicroStrategy achieves four-tier load balancing by incorporating load balancers into MicroStrategy Web.

© 2016, MicroStrategy Inc.

login ID

A text string usually entered along with a password during login; sometimes called a user name.

low watermark (LWM)

A value used by Intelligence Server to calculate the available memory for a new memory request for both virtual memory and virtual bytes. The low watermark is set as 95 percent of the high watermark.

See also **high watermark (HWM)**.

managed object

A managed object is a schema object unrelated to the project schema, which is created by the system and stored in a separate system folder. Managed objects are used to map data to attributes, metrics, hierarchies and other schema objects for Freeform SQL, Query Builder, and MDX cube reports.

matching cache

Report results retained for the purpose of being reused by the same report requests later on.

matching-history cache

A matching cache with at least one History List message referencing it.

memory request idle mode

The mode in which Intelligence Server denies requests for memory until its memory usage drops below the low watermark.

message lifetime

Determines how long (set in days) messages can exist in a user's History List.

metadata

A repository whose data associates the tables and columns of a data warehouse with user-defined attributes and facts to enable the mapping of the business view, terms, and needs to the underlying database structure. Metadata can reside on the same server as the data warehouse or on a different database server. It can even be held in a different RDBMS.

metadata synchronization

The process of synchronizing object caches across all nodes in a cluster.

node

Each machine in a cluster.

object browsing

The process of retrieving objects from the metadata by expanding or selecting a folder in Developer or MicroStrategy Web.

© 2016, MicroStrategy Inc.

object cache

A recently used object definition stored in memory on Developer and Intelligence Server.

ODBC

See Open Database Connectivity.

ODBC driver

A software routine that translates Intelligence Server requests into commands that the DBMS understands.

Open Database Connectivity (ODBC)

An open standard with which client computers can communicate with relational database servers. Client machines make a connection to a particular logical database, on a particular physical database server, using a particular ODBC driver.

orphan session

An entry in the statistics database that indicates that a session was initiated in Intelligence Server, but no information was recorded when the session ended.

password

Preserves user account integrity in an application. Many applications can associate both a password and a password hint with each user.

permissions

Define for objects the degree of control users have over them.

personal page execution

A type of Narrowcast Server implementation that executes one multi-page report for all users in a segment and then uses this single report to provide personalized content (pages) for different users. All users have their reports executed under the context of the same Intelligence Server user, so individual security profiles are not maintained. However, load on Intelligence Server may be significantly lower than for personalized report execution (PRE) in some cases.

See also **personal report execution**.

personal report execution

A type of Narrowcast Server implementation that executes a separate report for each set of users with unique personalization. Users may have reports executed under the context of the corresponding Intelligence Server user if desired. Using this option, security profiles defined in Developer are maintained. However, if there are many users who all have unique personalization, this option can place a large load on Intelligence Server.

See also **personalized page execution**.

physical warehouse schema

A detailed graphic representation of your business data as it is stored in the data warehouse. It organizes the logical data model in a method that makes sense from a

© 2016, MicroStrategy Inc.

database perspective.

See also **schema**.

private bytes

The current number of bytes a process has allocated that cannot be shared with other processes.

See also **virtual bytes**.

privilege

Defines what types of operations certain users and user groups can perform in the MicroStrategy system. For example, which objects a given user can create and which applications and editors he can use.

project

1) The MicroStrategy object in which you define all of the schema and application objects, which together provide for a flexible reporting environment. A project is the highest-level intersection of a data warehouse, metadata repository, and user community, containing reports, filters, metrics, and functions.

2) An object containing the definition of a project, as defined in (1). The project object is specified when requesting the establishment of a session.

project source

Defines a connection to the metadata database and is used by various MicroStrategy products to access projects. A direct project source is a two-tier connection directly to a metadata repository. A server project source is a three-tier connection to an Intelligence Server. One project source can contain many projects and the administration tools found at the project source level are used to monitor and administer all projects in the project source.

result cache

A result set from an executed report or document that is stored on Intelligence Server.

report cost

An arbitrary value you can assign to a report to help determine its priority in relation to other requests.

report instance

A container for all objects and information needed and produced during report execution including templates, filters, prompt answers, generated SQL, report results, and so on. It is the only object that is referenced when executing a report, being passed from one special server to another as execution progresses.

schedule

A MicroStrategy object that contains information specifying when a task is to be executed.

© 2016, MicroStrategy Inc.

scheduling

An Intelligence Server feature that is used to automate specific tasks.

schema

1) The set of tables in a data warehouse associated with a logical data model. The attribute and fact columns in those tables are considered part of the schema itself.

2) The layout or structure of a database system. In relational databases, the schema defines the tables, the fields in each table, and the relationships between fields and tables.

schema object

MicroStrategy object created, usually by a project designer, that relates the information in the logical data model and physical warehouse schema to the MicroStrategy environment. These objects are developed in MicroStrategy Architect, which can be accessed from Developer. Schema objects directly reflect the warehouse structure and include attributes, facts, functions, hierarchies, operators, partition mappings, tables, and transformations.

secure sockets layer (SSL)

An encryption technology that encodes the communication between a Web browser and Web server so that only the recipient can read it.

security filter

A qualification associated with a user or user group that is applied to all queries executed by that user or group.

security role

A MicroStrategy object that is used to store a particular grouping of privileges that you can apply to users or groups from project to project.

security view

A feature of most relational databases that restricts a user's access to the data so he can view only a subset of it.

server definition

An instance of Intelligence Server and all of its configuration settings.

server object

A configuration-level object in the metadata called server definition. It contains governing settings that apply at the server level, a list of projects registered on the server, connection information to the metadata repository, and so on.

server state dump (SSD)

A collection of information related to the current state of Intelligence Server that is written to a log file.

© 2016, MicroStrategy Inc.

system prompt

A special type of prompt that does not require an answer from the user. A system prompt is answered automatically by the system. For example, the User Login system prompt is answered automatically with the login name of the user who runs the report. System prompts can be used in filters and metric expressions.

task list

A list of tasks that must be accomplished to complete a job within Intelligence Server.

update package

A file containing object definitions and conflict resolution rules for those objects. Update packages are created using Object Manager. When an update package is applied to a project or project source, the objects are copied into the project or project source. If the objects already exist, the specified conflict resolution rules are followed.

used dependency

Occurs when an object uses other objects in its definition.

See also **used-by dependency**.

used-by dependency

Occurs when an object is used as part of the definition of other objects.

See also **used dependency**.

user

A person who can log in to a MicroStrategy system, create and own objects such as reports, execute reports, and take advantage of all the other features in the system.

user address space

Sometimes referred to as virtual address space. Independent of virtual memory and of finite size. It is measured per process on the machine (such as the MSTRSVR.exe Intelligence Server application). By definition, in a 32 bit operating system, virtual bytes is limited to 4GB (232). By default, Windows operating system divides this into two parts UAS and System Address Space (SAS). The UAS is, in this case, for Intelligence Server to store data and code while the SAS is for the operating system's use.

user group

"Group" for short. A collection of users.

user profile

What the user is doing when he or she is logged in to the system.

user session

Established when each user connects to Intelligence Server from a MicroStrategy client (MicroStrategy Web, Developer, MicroStrategy Mobile, and so on).

© 2016, MicroStrategy Inc.

virtual bytes

The limit associated with Intelligence Server's virtual address space allocation is the committed address space (memory actually being used by a process) plus the reserved address space (memory reserved for potential use by a process).

See also **private bytes**.

virtual memory

The amount of physical memory (RAM) plus Disk Page file (also called the swap file).

VLDB property

A group of settings used to control SQL syntax or behavior for different DBMS platforms. VLDB properties initialize the SQL generation standards for each DBMS platform and allow you to optimize SQL generation for your data warehouse configuration.

VLDB settings

Settings that affect the way Intelligence Server interacts with the data warehouse to take advantage of the unique optimizations that different databases offer. Each VLDB property has two or more of VLDB settings.

working set

A collection of messages that reference in-memory report instances. A message is added to the working set when a user executes a report or retrieves a message from his or her Inbox.

XML cache

A report cache in XML format that is created and available for use on the Web.

© 2016, MicroStrategy Inc.

INDEX

6

6.X database authentication mode *105*

A

access control

- application functionality *58*
- to data via the database *91*

Access Control List *59*

- assigning *58*
- controlling application access *64*
- Object Manager *247*
- User Merge Wizard *95*
- viewing *55*

accessing

- Contact Editor *524*
- Device Editor *514*
- Transmitter Editor *507*

ACL. See Access Control List. *59*

active user *324*

addresses for contacts *527*

administration tasks, scheduling *485*

Administrator page in MicroStrategy Web *531*

- accessing *531*

Amazon cloud

- creating an Amazon Machine Image *626*
- launching *628*
- managing *629*
- retrieving status *627*
- terminating *629*

anonymous authentication mode *102*

Apache Hive Wire Protocol

- ODBC Driver for *598*

© 2016, MicroStrategy Inc.

application development life cycle *220*
developer *222*
recommended *220*
application object *5*
copying *234*
applying project defaults for MicroStrategy Web *533*
architecture overview *1*
archiving History List messages *434*
assigning
permissions *58*
privileges individually *67*
privileges with security role *69*
auditing for license compliance *208*
authentication *97*
example *180*
implementing Windows *133*
LDAP *105*
LDAP binding *117*
LDAP database passthrough *117*
LDAP password comparison *117*
standard *101*
troubleshooting *756*
authentication mode *98*
6.X database *105*
anonymous *102*
automating a process with Command Manager *651*

B

baseline-versus-baseline integrity test *667*
baseline-versus-project integrity test *667*
best practices
cache management *402*
contacts *523*
bursting file subscriptions
into multiple files *496*
into subfolders *497*

C

cache *399*
best practices *402*
centralized cache file *376*
database connection *9*
element. See element cache. *440*
in a cluster *395*
object. See object cache. *451*
purging for a project *26*
purging for a server *27*
result cache types *410*
sharing *376*
types *399*
viewing *409*
Cache Monitor *409*
callstack *749*
canceling a job *48*

© 2016, MicroStrategy Inc.

centralized cache file *376*
certificate
installing for SSL *188*
change journaling *25, 281*
comments *283*
enabling *281*
exporting entries *286*
filtering entries *285*
logged information *281*
project *281*
project source *281*
purging *286*
searching entries *285*
viewing entries *283*
changing project status *30*
child dependency
locating *238*
cloud-based environment administration *625*
creating a vApp *630*
creating a virtual machine *635*
creating an Amazon Machine Image *626*
deleting a vApp *633*
duplicating a vApp *633*
launching an Amazon cloud environment *628*
managing an Amazon cloud environment *629*
restarting a vApp *631*
restarting a virtual machine *632*
retrieving Amazon cloud status *627*
starting a vApp *631*
starting a virtual machine *632*
stopping a vApp *631*
stopping a virtual machine *632*
terminating an Amazon cloud environment *629*
cluster *370*
See also clustering. *370*
Cluster Monitor *26*
accessing *26*
using *26*
Cluster view
accessing *26*
System Maintenance Monitor *26*
using *26*
clustering *369*
architecture *372*
backing up caches *417*
benefits *370*
caches *395*
components *370*
Intelligence Server machines *386*
monitoring *26*
requirements *370*
sharing a History List across a cluster *377*
sharing information between nodes *373*

© 2016, MicroStrategy Inc.

sharing metadata across a cluster *373*

Web *370*

combining users *93*

comma-separated value file for export *45*

command line

Integrity Manager *673*

Integrity Manager exit codes *675*

Command Manager *651*

encrypting scripts *659*

execution errors *660*

privileges *652*

scheduling with Windows AT command *652*

syntax errors *660*

timeout errors *661*

comment in a workflow *567*

communications privacy *183*

comparing reports *665*

baseline-versus-baseline comparison *667*

baseline-versus-project comparison *667*

project-versus-project comparison *667*

viewing differences *684*

complying with license *208*

concurrent user *324*

configuration in MicroStrategy System Manager *545*

configuration object *5*

copying *234*

moving *234*

Configuration Wizard

and Health Center *702-703*

Configuration Wizard response file *722*

configuring

Health Agent *703*

JAAS *154*

Master Health Agent *702*

system architecture *306*

conflict resolution default action *246*

connection borrowing *340*

connection mapping *74*

example *181*

purpose *74*

User Merge Wizard *96*

connection pooling *115*

connector *553*

Contact Editor, accessing *524*

contact group *526*

contacts

about *503*

address maintenance *527*

best practices *523*

Contact Editor *524*

copying *524*

creating *522*

© 2016, MicroStrategy Inc.

deleting *524*, *528*

disabling *524*

duplicating *524*

enabling *524*

example *503*, *522*

linked user maintenance *527*

prerequisites *522*

recommended tasks *523*

renaming *524*

searching for dependent subscriptions *524*

cookie *537*

copying

application object *234*

configuration object *234*

object *230*

project *220*, *225*

resolving a conflict when *243*

schema object *234*

CPU affinity *210*

for MicroStrategy Web *214*

CPU license

managing *206*

verifying *205*

creating

contact *522*

contact group *526*

device *512*

group *56*

integrity test *670*

log file *744*

schedule *479*

transmitter *508*

user *56*

CSV, exporting as *45*

Cube Monitor *458*

D

data access

connection mapping *74*

passthrough execution *77*

security filter *79*

security using RDBMS *91*

security view *91*

splitting a fact table by column *93*

splitting a fact table by row *91*

data security *73*

data source name *595*

MicroStrategy System Manager *595*

data warehouse *4*

authentication by *103*

connection mapping *74*

connection thread *337*

DBMS upgrading *11*

design and tuning *307*

leveraging the security view *75*

limiting connection threads *753*

© 2016, MicroStrategy Inc.

monitoring connection to *9*

navigating in *4*

optimizing connection threads *338*

performance *306*

placement *308*

platform *306*

setting optimal number of connection threads *337*

SQL maximum size *336*

SQL timeout per pass *336*

timestamp *294*

troubleshooting *731*

data warehouse connection, governing *338*

database authentication mode *103*

database connection

busy *9*

cache *9*

deleting *10*

Enterprise Manager *290*

governing summary *365*

information in the SSD *750*

restricting element caching *449*

viewing current *10*

database connection cache *9*

governing *339*

Database Connection Monitor *9*

database connection thread

governing *338*

optimizing *338*

prioritizing *340*

timeout *339*

dataset *41*

DB2 for i ODBC driver *596*

DB2 for UDB iSeries ODBC driver *596*

DB2 UDB ODBC driver (UNIX/Linux) *595*

DB2 z/OS ODBC driver *597*

decision process *555*

default location for the Web log file *745*

default options for MicroStrategy Web *533*

applying *533*

loading *533*

user preferences *533*

deleting

database connection *10*

device *521*

element cache *450*

History List messages *437*

object cache *454*

report cache *410*

transmitter *512*

user *56*

deployment, sizing *310*

designing a Narrowcast Server application *367*

© 2016, MicroStrategy Inc.

device
about *503*
best practices *514*
creating *512*
deleting *521*
Device Editor *514*
environment *503*
example *512*
recommended tasks *513*
types *512*
Device Editor *514*
diagnosing system problems using Health Center *708*
diagnostics *731*
configuring *734*
configuring a performance counter *736*
log file anatomy *745*
sample log file *746*
Diagnostics and Performance Logging tool *299*, *732*
diagnostics package *711*
creating in Health Center *711*
sending to technical support *711*
disabling
change journaling comments *283*
contacts *524*
document caching *408*
element caching *442*
object caching *452*
report caching *408*
single report caching *424*
disconnecting
database *9*
user *57*
Distinguished Name (DN) *120*
troubleshooting LDAP *759*
Distribution Services *501*
document *41*
instance *41*
system performance and *354*
testing execution of *667*
document cache
file format *405*
index file *405*
managing *412*
monitoring *409*
viewing *409*
document instance *41*
double-byte characters *662*
drill path, XML *66*
duplicating a project *220*, *225*
dynamic recipient list *493*

E

element browsing *38*
job processing steps *38*
element cache *440*
deleting *450*
disabling *442*

© 2016, MicroStrategy Inc.

enabling *442*

limiting attribute elements *448*

limiting memory usage *446*

limiting number of elements *443*

matching algorithm *442*

optimizing *445*

pool *441*

restricting by database connection *449*

restricting by database login *449*

settings *451*

storage location *442*

terminology *441*

element incremental fetch

optimizing *445*

setting *443*

element request *441*

enabling

contacts *524*

element caching *442*

single report caching *424*

encryption *538*, *659*

metadata *4*

Enterprise Manager

database connection *290*

log statistics *295*

statistics troubleshooting *777*

system check in Health Center *716*

entry process *554*

event-based schedule *480*

event triggering

automatic *484*

manual *485*

example

authentication *180*

connection mapping *181*

contacts *503*, *522*

fact table partitioning *181*

MCM *322*

memory depletion *322*

partitioned fact table *181*

passthrough execution *180*

project and data warehouse maintenance scenarios *31*

standard authentication *181*

Windows authentication *180*

Excel file with formatting, exporting as *45*

exception *747*

execute permission *65*

executing a job *32*

canceling *48*

current job *47*

governing *333*

governing by project *333*

governing by user *333*

prioritizing *340*

seeing the current job *47*

© 2016, MicroStrategy Inc.

SSD information *750*
viewing the current job *47*
executing an integrity test *671*
command line *673*
Integrity Manager Wizard *671*
execution idle project mode *28*
execution time, governing *334*
exit code *637*
exit process *559*
expiring a report cache *416*
export size, governing *350*
exporting
change journal entries *286*
CSV *45*
Excel file with formatting *45*
PDF *46*
plain text *45*
report from Web *44*, *350*

F

fact table partitioning
example *181*
fact table partitioning as a security measure *91*
example *181*
failover *371*
failover support *371*
fatal exception *747*
filtering
change journal *285*
History List messages *440*
firewall *534*
Freeform SQL
deleting managed objects *276*
full idle project mode *29*

G

governing
database connection cache *339*
database connection thread *338*
execution time *334*
export size *350*
History List *327*
jobs per project *333*
jobs per user *333*
memory for Web requests *323*
number of jobs *332-333*
report size *343*
request *331*
results delivery *347*
settings *355*
SQL *335*
user profile *329*
user resources *327*
user session *325*
user session recovery (Web) *329*
working set *328*
XML cell *349*
XML drill path *351*

© 2016, MicroStrategy Inc.

graph in Integrity Manager *685*
Greenplum Wire Protocol
ODBC Driver for *598*
group *52*
creating *56*
default *52*
importing *56*
importing an LDAP group *127*
LDAP search filter *114*
linking LDAP groups *120*
merging *93*
Guest user group *53*

H

Health Agent *703*
adding to Health Center system *704*
configuring *703*
finding (discovering) *704*
firewall and *706*
Health Center *699*
accessing *700*
adding Health Agents to *704*
console *700*
diagnostic package *711*
Enterprise Manager system check *716*
finding (discovering) Health Agents *704*
fixing problems found by *709*
Health Agent *703*
Health Agent configuration *703*
Intelligence Server system check *716*
log file in *710*
manual system check *708*
Master Health Agent *702*
Master Health Agent configuration *702*
MicroStrategy Web system check *721*
Narrowcast Server system check *720*
network *700*
scheduled system check *708*
system check *713*
Health Center console *700*
high watermark (HWM) *319*
history cache *404*
Cache Monitor *409*
History List *425*
accessing *432*
archiving *434*
backing up *436*
caching and *428*
clustered environment and *436*
deleting messages *437*
governing *327*
lifetime *437*
managing *435*
migration *581*
repository *579*

© 2016, MicroStrategy Inc.

sharing across a cluster *377*

History List message

deleting *437*

filtering *440*

purging *440*

viewing details *440*

HTML document *42*

instance *43*

job processing steps *42*

HTML document instance *43*

HWM. See high watermark. *319*

I

idle time *326*

idling a project *26*, *30*

IIS tuning *544*

Import Data, deleting managed objects *276*

importing

group *56*

user *56*

user group *56*

inbox synchronization *377*

inbox, SSD information *750*

incremental fetch in Web *348*

index file

document cache *405*

report cache *405*

Informix Wire Protocol

ODBC Driver for *599*

Informix XPS

ODBC Driver for *600*

inheriting privileges *69*

installation, silent *48*

Integrated authentication, linking login to LDAP user login *142*

Integrity Manager *665*

creating an integrity test *670*

graph *685*

integrity test creation *670*

loading saved test settings *671*

prerequisites for using *666*

prompted report and *676*

report data *684*

report execution output *686*

saving test settings *671*

test results, viewing *683*

viewing differences in reports *684*

viewing saved results of a test *686-687*

viewing test results *683*

Integrity Manager Wizard *666*, *671*

integrity test *666*

baseline-versus-baseline *667*

baseline-versus-project *667*

creating *670*

executing from command line *673*

executing with Integrity Manager Wizard *671*

© 2016, MicroStrategy Inc.

loading *672*
project-versus-project *667*
report status *683*
saving *671*
single-project *667*
viewing results of *683*
viewing saved results *686-687*
Intelligence Server
component, MCM *316*
components *581*
components information in the SSD *750*
configuring *580*
CPU type and speed *311*
CPUs, number of *311*
diagnostics *731*
governing *356*
job processing *32*
logging *732*
number of CPUs *311*
request types *32*
running mode *14*
runtime memory use *315*
SSD components information *750*
SSD version information *748*
starting *12*
as a service *15*
as an application *19*
automatically *17*
troubleshooting *731*
startup memory use *314*
stopping *13*
system check in Health Center *716*
thread balancing *343*
troubleshooting *731*
unregistering as a service *14*
version in SSD *748*
Intelligent Cube *457*
Cube Monitor *458*
job processing steps *36*
viewing *458*
invalidating a report cache *410*, *413*

J

JAAS. See Java Authentication and Authorization Service. *154*
Java Authentication and Authorization Service *154*
job *32*
execution *337*
governing *332*
Job Monitor *47*
viewing currently executing job *47*
job priority *340*
setting *341*
variables *340*
job processing steps *32*
element browsing *38*

© 2016, MicroStrategy Inc.

graph report *37*
HTML document *42*
Intelligent Cube report *36*
object browsing *37*
report *33*
report with a prompt *36*
Web *44*
Web exporting *44*
JVM heap size *544*

L

latency and project failover *392*
LDAP *105*
anonymous/guest user *121*
attribute
controlling project access with *131*
importing *131*
in a security filter *131*
authentication. See LDAP authentication. *105*
connecting to a LDAP server with the LDAP SDK *110*
connection pooling *115*
clustered LDAP servers *115*
directory *105*
group search filter *114*
importing
at login *127*
in batch *127*
integrated authentication credentials *142*
LDAP attribute *131*
on a schedule *130*
users and groups *127*
importing vs. linking users *118*
information flow *106*
linking
integrated authentication user *142*
users and groups *120*
Windows login *137*
linking vs. importing users *118*
requirements for connecting to MicroStrategy *107*
scheduling user import and synchronization *130*
SDK *110*
UNIX/Linux *112*
search filter *112*
search root *113*
security settings after importing *129*
server *105*
SSL certificate *124*
synchronizing user and group information *121*
synchronizing users and groups
on a schedule *130*
troubleshooting *132*
user and group synchronization *121*

© 2016, MicroStrategy Inc.

user group *53*

user privileges after importing *129*

user search filter *114*

verifying and importing users and groups at login *112*

LDAP authentication *105*

binding *117*

database passthrough *117*

FAQ *761*

managing *125*

password comparison *117*

setting up *123*

settings *126*

troubleshooting *757*

license

auditing *208*

check *205*

compliance and CPU affinity *211*

CPU *205*

managing *206*

out of compliance *206*

time to run license check *205*

updating *209*

verifying named user *204*

License Manager *207*

Lightweight Directory Access Protocol. See LDAP. *105*

Linux, install LDAP SDK *112*

load balancing *371*

loaded project mode *27*

loading

project *26*, *30*

project defaults for MicroStrategy Web *533*

report cache from disk *410*

saved report cache *413*

local cache file *374*

locking a project *231*

log file

creating *744*

diagnostics *745*

Health Center and *710*

location *745*

reading *745*

viewing *745*

logging configuration *732*

logging subscriptions statistics *779*

login ID, using to restrict element caching *449*

low watermark (LWM) *319*

LWM. See low watermark. *319*

M

machine size recommendations *310*

managed object *276*

deleting from a project *277*

managing

license *206*

object individually *232*

© 2016, MicroStrategy Inc.

project configuration *25*
project security *25*
project status *25*
Master Health Agent *702*
best practices for selecting *702*
configuring *702*
matching-history cache *404*
matching caches *403*
maximum single allocation size *317*
MCM
criteria for granting a request *318*
example *322*
governing Intelligence Server memory use *316*
granting or denying memory requests *318*
logic flow *318*
managing virtual address space fragmentation *321*
maximum use of virtual address space *320*
memory request idle time *318*, *321*
memory watermark *320*
minimum reserved memory *317*, *320*
preventing memory depletion *750*
single memory allocation governing *317*
tasks governed *316*
using *317*
XML and *316*
MDX cube source, deleting managed object *276*
memory *312*, *751*
governing for Web requests *323*
SSD information *748*
top consumers *752*
Memory Contract Manager. See MCM. *316*
memory depletion *750*
example *322*
preventing *750*
SSD information *749*
memory request idle mode *321*
memory request idle time *318*
recommended value *321*
merging
groups *93*
projects *266*
users *93*
message lifetime *437*
metadata *4*
connecting to *7*
fixing inconsistencies in *762*
sharing across a cluster *373*
synchronization *373*
metadata repository *579*
creating *579*
metadata synchronization *373*

© 2016, MicroStrategy Inc.

Microsoft Access

connecting as data source (UNIX/Linux) *602*

Microsoft SQL Azure

ODBC driver *600*

Microsoft SQL Server

ODBC driver *600*

MicroStrategy Application Deployment Manager, creating a metadata repository *579*

MicroStrategy applications

configuring for SSL *190*

MicroStrategy Distribution Services

migrating subscriptions from Narrowcast Server *581*

MicroStrategy Health Center. See Health Center. *699*

MicroStrategy Integrity Manager *665*

MicroStrategy Mobile

firewall *534*

managing *529*

security *534*

MicroStrategy Narrowcast Server

migrating subscriptions to Distribution Services *581*

MicroStrategy Office, installing from Web *542*

MicroStrategy Project Comparison Wizard *272*

MicroStrategy Project Merge *266*

MicroStrategy System Manager *545*

comment *567*

connector *553*

decision *555*

entry process *554*

exit process *559*

iterative retrieval process *560*

parameters *638*

process *578*

workflow

creating *546*

deploying *644*

sample *571*

template *571*

validating *568*

MicroStrategy user group. See user group. *52*

MicroStrategy User Merge Wizard *93*

MicroStrategy user. See user. *52*

MicroStrategy Web *745*

Administrator page *531*

cookie *537*

CPU affinity feature *214*

default options *533*

edition *530*

encryption *538*

file-level security *539*

firewall *534*

© 2016, MicroStrategy Inc.

Flash Mode *530*

log file *745*

managing *529*

Narrowcast Server and *541*

security *534*

system check in Health Center *721*

user settings *533*

MicroStrategy Web Universal, security roles with *532*

migrating an object *230*

minimum reserved memory *317*

monitoring

cluster *26*

database connection *9*

document cache *409*

job *47*

machine performance *311*

report cache *409*

user connection *57*

moving a configuration object *234*

MySQL

Wire Protocol driver for *603*

N

named user license

managing *206*

setting when to verify *205*

verifying *204*

Narrowcast Server

and deleted user *56*

application design *367*

connecting to Intelligence Server *368*

information source *368*

integrating with MicroStrategy Web *541*

system resource use and *367*

Narrowcast Server system check in Health Center *720*

network

bandwidth *309*

monitoring *309*

performance considerations *307*

recommended configuration *308*

network address of the temp client *47*, *57*

node *370*

notification for failed transmission *507*

O

object browsing *37*

job processing steps *37*

object cache *451*

deleting *454*

disabling *452*

limiting memory usage *452*

matching algorithm *452*

summary of settings *455*

© 2016, MicroStrategy Inc.

Object Manager *232*
Access Control List *247*
copying an object *230*
international settings *242*
Project Merge versus *231*
resolving a conflict *243*
saving a dependent object *240*
starting *236*
table migration *248*
timestamp *242*
used-by dependency *239*
used dependency *241*
ODBC driver *600*
Apache Hive Wire Protocol *598*
DB2 for i *596*
DB2 for UDB iSeries *596*
DB2 UDB (UNIX/Linux) *595*
DB2 z/OS *597*
Greenplum Wire Protocol *598*
Informix Wire Protocol *599*
Informix XPS *600*
Microsoft SQL Azure *600*
Microsoft SQL Server *600*
MySQL Wire Protocol *603*
Oracle Wire Protocol *603*
PostgreSQL Wire Protocol *605*
Salesforce *605*
Sequelink *602*
Sybase ASE Wire Protocol *606*
OLAP Services and system performance *346*, *355*
Oracle 8i *12*, *765*
Oracle Wire Protocol
ODBC driver for *603*

P

page-by and system performance *354*
parameters
commenting in file *640*
exporting for System Manager *640*
importing for System Manager *640*
System Manager workflows *638*
partial idle project mode *29*
partitioned fact table *91*
example *181*
passthrough execution *77*
example *180*
PDF, exporting as *46*
performance counters and diagnostics *736*
Performance Monitor *311*
permissions *58*
assigning *58*
list of *61*
server object *62*
use and execute *65*
viewing *55*
personal page execution *367*

© 2016, MicroStrategy Inc.

personal report execution *367*
personalized drill path *66*
physical disk utilization *312*
physical warehouse schema *4*
plain text, exporting as *45*
PostgreSQL Wire Protocol
ODBC driver for *605*
preferences, Web *533*
primary server *25*
prioritizing database connection thread *340*
privacy of communications *183*
private bytes *312*
privileges *66*
assigning *67*
denying *67*
governing with *329*
granting *67*
inheriting *69*
merging in User Merge *94*
process *578*
administering cloud-based environment *625*
administering MicroStrategy services *586*
automating administrative tasks *589*
completing a separate System Manager workflow *607*
configuring MicroStrategy components *579*
configuring Intelligence Server *580*
creating a project source *580*
creating repository tables *579*
migrating History List repositories *581*
migrating subscriptions to Distribution Services *581*
upgrading Intelligence Server components *581*
upgrading statistics repositories *581*
creating *547*
creating data source names *595*
decision *555*
entry *554*
exit *559*
exit code *637*
iterative retrieval *560*
managing a project *582*
parameter *638*
resolving *637*
system *609*
verifying reports and documents *592*
processing results *343*
processor affinity *210*
project *6*
cache viewing *409*

© 2016, MicroStrategy Inc.

change journaling *25*, *281*

changing status of *30*

comparing *272*

comparing reports between *667*

copying *220*, *225*

default options for MicroStrategy Web *533*

deleting change journal entries *286*

duplicating *220*, *225*

enabling change journaling *281*

governing summary *361*

idling *26*, *30*

loading *26*, *30*

locking *231*

logging changes *281*

merging *266*

MicroStrategy Web default options *533*

purging a cache *26*

resuming *26*, *30*

SSD information *748-749*

state information in the SSD *749*

testing *665*

tracking *274*

unloading *26*, *30*

unlocking *231*

viewing changes *283*

project-versus-project integrity test *667*

Project Comparison Wizard *272*

comparing objects between two projects *273*

project comparison process *273*

using *273*

project failover support *392*

Project Merge *266*

command *268*

conflict resolution *271*

Object Manager versus *231*

project merge process *267*

resolving a conflict *271*

scheduling with Windows AT command *271*

using *268*

Project Merge Wizard *267*

conflict resolution *271*

merging projects *268*

resolving a conflict *271*

XML *231*

project mode *27*

execution idle *28*

full idle *29*

loaded *27*

partial idle *29*

request idle *28*

unloaded *27*

warehouse execution idle *29*

Project Monitor *25*

accessing *25*

© 2016, MicroStrategy Inc.

scheduling maintenance *26*

using *25*

project schema

exporting *26*

printing *26*

project source *6*, *580*

change journal

disabling *281*

enabling *281*

purging *287*

change journaling *281*

project status *27*

changing *30*

Project view

accessing *25*

using *25*

Project view of System Maintenance Monitor *25*

prompt

Command Manager and *661*

Integrity Manager and *676*

resolving *36*

system *89*

system performance and *354*

prompted report in Integrity Manager *676*

Public user group *53*

purging

change journal *286*

change journal for a project source *287*

History List messages *440*

result caches *415*

Q

Query Builder, deleting managed objects *276*

R

report

comparing

from two baselines *667*

with a baseline *667*

with another report *665*

cost *342*

instance *33*

prompted in Integrity Manager *676*

scheduling *488*

size *343*

system performance and *353*

testing execution of *667*

report cache

configuring settings *416*

deleting *410*, *414*

disabling for

a project *408*

a single report *424*

enabling for a single report *424*

expiring *416*

© 2016, MicroStrategy Inc.

file format *405*

history *404*

index file *405*

invalidating *410*, *413*

loading from disk *410*, *413*

location *404*

managing *412*

matching *403*

matching-history *404*

matching algorithm *406*

memory usage *421*

monitoring *409*

project settings *417*

scheduling an update *412*

settings *416*

status *410*

types *403*, *410*

unloading to disk *410*, *413*

updating on a schedule *488*

viewing *409*

XML *404*

Report Cache Monitor *410*

report comparison tool *665*

report complexity and system resource use *353*

report cost *342*

report instance *33*

repository

History List *579*

metadata *579*

statistics *579*

request

governing *331*

types *32*

request idle project mode *28*

resolving a prompt *36*

response file *722*

Configuration Wizard (UNIX) *724*

Configuration Wizard (Windows) *724*

configuring *48*

installing *48*

result cache *400*

purging *415*

sharing options in cluster *374*

result row limit troubleshooting *769*

results

delivery *347*

processing *343*

results summary of integrity tests *683*

resuming a project *26*, *30*

S

Salesforce

ODBC driver for *605*

sample workflow *571*

Sarbanes-Oxley change journaling *281*

© 2016, MicroStrategy Inc.

Scan MD tool *765*
schedule *479*
creating *479*
event-based *480*
SSD information *749*
time-triggered *480*
scheduled e-mail, enabling with Web *541*
scheduling *477*
administration tasks *485*
Command Manager script *652*
project maintenance in Project Monitor *26*
Project Merge to run *271*
report *488*
schema object *5*
copying *234*
script *651*
execution errors *660*
syntax errors *660*
timeout errors *661*
Search Export feature *274*
security
assigning privileges *67*
authentication *98*
controlling application functionality *58*
cookies *537*
data access control *73*
encryption *538*
file-level *539*
firewall *534*
MicroStrategy Mobile *534*
MicroStrategy Web *534*
permissions *58*
privileges *66*
security role *69*
setting up *51*
single sign-on *52*
security filter *79*
LDAP attribute in *131*
managing with Project Monitor *25*
merging multiple *86*
metric level *82*
range attributes *82*
User Merge Wizard *96*
security role *69*
user group and *68*
User Merge Wizard *95*
security view *91*
purpose *91*
send now, enabling with Web *541*
Sequelink
ODBC driver for *602*
server definition *5*
Intelligence Server startup *12*
permissions *62*
SSD information *748*

© 2016, MicroStrategy Inc.

server object *62*
server state dump (SSD) *747*
contents *747*
server, purging a cache in *27*
Service Manager *15*
Service Principal Name (SPN)
application server *151*
IIS *149*
Intelligence Server *143*
setting
job priority *341*
security *51*
sharing caches *374, 376*
silent installation *48*
Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO) *155*
single-project integrity test *667*
single memory allocation governing *317*
single sign-on *52*
SPN. See Service Principal Name. *143*
SPNEGO *155*
SQL, limiting report's SQL size *335*
SSD. See server state dump. *747*
SSL
about *183*
certificate *124*
configuring with MicroStrategy applications *190*
installing a certificate *188*
testing *194*
troubleshooting in LDAP *758*
standard authentication *101*
example *181*
starting
Intelligence Server *12*
Intelligence Server automatically *17*
statistics
creating statistics tables *294*
options *288*
setting the project to log *295*
timestamp *294*
troubleshooting *777*
statistics repository *579*
upgrading *581*
status
cache *410*
project *27*
stopping Intelligence Server *13*
subscription
migrating from Narrowcast Server to Distribution Services *581*
subscriptions statistics logging *779*
Sybase ASE Wire Protocol
ODBC driver for *606*
synchronizing schedules for LDAP *130*

© 2016, MicroStrategy Inc.

System Administration Monitor *24*
system architecture *1*
configuring *306*
system check *708*, *713*
manual *708*
scheduled *708*
System Maintenance Monitor
Cluster view *26*
Project Monitor *25*
Project view *25*
system prompt *89*
system resources governing *310*

T

table
migrating with Object Manager *248*
partitioning *91*
task list *33*
temp client *47*, *57*
template workflow *571*
testing
document execution *667*
project *665*
report execution *667*
testing, SSL *194*
time-triggered schedule *480*
timestamp *294*
migrated object *242*
transmitter *502*
best practices *507*
deleting *512*
email *508*
failure notification *507*
file *509*
FTP *510*
iPad *510*
iPhone *511*
print *509*
recommended tasks *507*
Transmitter Editor *507*
types *508*
Transmitter Editor *507*
triggering an event
automatically *484*
manually *485*
troubleshooting *730*
LDAP *132*
report result rows *769*
tuning *301-302*
runtime capacity variables *304*
system design *303*
system requirements *302*

U

UNIX, install LDAP SDK *112*
unknown exception *747*
unloaded project mode *27*

© 2016, MicroStrategy Inc.

unloading
project *26*, *30*
report cache to disk *410*, *413*
unlocking a project *231*
update package *250*
creating *586*
System Manager workflow *585*
updating a license *209*
upgrading
VLDB properties *11*
use permission *65*
used-by dependency *239*
used dependency *238*
Object Manager *241*
user *52*
anonymous/guest for LDAP *121*
creating *56*
deleting *56*
disconnecting *57*
importing *56*
importing an LDAP user *127*
LDAP search filter *114*
linking LDAP users *120*
managing with Project Monitor *25*
merging *93*
profile *329*
resources *327*
sharing an account *100*
user address space *751*
User Connection Monitor *57*
viewing active user connections *57*
user connection, viewing *57*
user group *52*
default *52*
Guest *53*
importing *56*
LDAP *53*
Public *53*
security role and *68*
User Merge Wizard *93*
ACL *95*
connection mapping *96*
group membership *95*
privileges *94*
process *94*
profile folder *95*
running *96*
security filter *96*
security role *95*
user mode process dump *747*
user preferences, Web *533*
user session *324*
governing *325*
information in the SSD *750*
recovering (Web) *329*

© 2016, MicroStrategy Inc.

V

validating a workflow *568*
manual *568*
vCloud
creating a vApp *630*
creating a virtual machine *635*
deleting a vApp *633*
duplicating a vApp *633*
restarting a vApp *631*
restarting a virtual machine *632*
starting a vApp *631*
starting a virtual machine *632*
stopping a vApp *631*
stopping a virtual machine *632*
verifying a named user license *204*
viewing
change journal entries *283*
current database connection *10*
History List message details *440*
log file *745*
log file in Health Center *710*
results of an integrity test *683*
virtual bytes *751*
virtual memory *312*
VLDB properties *11*
governing summary *366*
upgrading *11*

W

warehouse execution idle project mode *29*
Windows AT command
Command Manager *652*
Project Merge *271*
Windows authentication
example *180*
implementing *133*
troubleshooting *756*
Windows Performance Monitor *311*
memory *313*
Windows, linking login to LDAP user login *137*
workflow
comment *567*
connector *553*
creating *546*
decision process *555*
deploying *644*
deploying with System Manager *644*
deploying with the command line *647*
entry process *554*
exit process *559*
iterative retrieval process *560*
sample *571*
template *571*

© 2016, MicroStrategy Inc.

testing with System Manager *644*

validating *568*

working set *328*

governing *328*

X

XML

cache *404*

converting to HTML for Web *44*

delta and reports *328*

drill path

governing *351*

personalized *66*

MCM and *316*

Project Merge Wizard *231*

results delivery and *347*

size *349*

XML cache *404*

Cache Monitor *409*

© 2016, MicroStrategy Inc.