Nineteenth Edition, June 2011, version 9.2.1

To ensure that you are using the document that corresponds to the software you are licensed to use, compare this version number with the software version shown in “About MicroStrategy...” in the Help menu of your software.

Document number: 09470921
Copyright © 2011 by MicroStrategy Incorporated. All rights reserved.

If you have not executed a written or electronic agreement with MicroStrategy or any authorized MicroStrategy distributor, 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 © 2001-2011 by MicroStrategy Incorporated. All rights reserved.

THIS DOCUMENT 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 NON-INFRINGEMENT, 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, Inc. 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 any case, 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, Inc., 1850 Towers Crescent Plaza, Vienna, VA 22182. Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.


All other company and product names may be trademarks of the respective companies with which they are associated. 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. 6,154,766, 6,173,310, 6,260,050, 6,263,051, 6,269,393, 6,279,033, 6,567,796, 6,587,547, 6,606,596, 6,658,093, 6,658,432, 6,692,195, 6,671,715, 6,691,100, 6,694,316, 6,697,808, 6,704,723, 6,741,980, 6,765,997, 6,788,788, 6,722,137, 6,788,768, 6,788,987, 6,801,910, 6,820,073, 6,836,537, 6,859,798, 6,873,693, 6,885,734, 6,940,953, 6,964,012, 6,977,992, 6,996,588, 6,996,569, 7,003,512, 7,010,518, 7,016,480, 7,020,251, 7,039,165, 7,082,422, 7,113,993, 7,127,403, 7,174,349, 7,181,417, 7,194,457, 7,197,481, 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, 7,440,896, 7,446,780, 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, and 7,945,584. Other patent applications are pending.

Various trademarks and/or registered trademarks of MicroStrategy products are copyright the technology of third parties. This product may contain one or more of the following copyright technologies:

Graph Generation Engine Copyright © 1998-2011. Three D Graphics, Inc. All rights reserved.
Actuate® Formula One. Copyright © 1993-2011 Actuate Corporation. All rights reserved.
XML parser Copyright © 2003-2011 Microsoft Corporation. All rights reserved.
Xalan XSLT processor. Copyright © 1998-2011, The Apache Software Foundation. All rights reserved.
Xerces XML parser. Copyright © 1999-2011, The Apache Software Foundation. All rights reserved.
FOP XSLT formatting objects. Copyright © 2004-2011, The Apache Software Foundation. All rights reserved.
Portions of Intelligence Server memory management Copyright © 1991-2011 Compuware Corporation. All rights reserved.
ASIHTTPRequest library. Copyright © 2007-2011, All-Seeing Interactive. All rights reserved.
This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)
CONTENTS

Description of Guide ................................................................. xix
About this book .......................................................................... xxiii
    Additional formats ............................................................... xxiii
    How to find business scenarios and examples ...................... xxiii
    What’s new in this guide .................................................... xxiv
    Who should use this guide .................................................. xxvi

Resources .................................................................................... xxvii
    Documentation ........................................................................ xxvii
    Education ............................................................................. xxxvi
    Consulting ............................................................................ xxxvi
    International support ........................................................ xxxvi
    Technical Support .............................................................. xxxvii

Feedback ..................................................................................... xlii

1. Introduction to MicroStrategy System Administration

Introduction .................................................................................. 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 ........................... 6
        Tying it all together: projects and project sources ............... 7
    Communicating with databases ............................................. 8
        Connecting to the MicroStrategy metadata ......................... 8
        Connecting to the data warehouse .................................... 9
        Caching database connections ....................................... 10
        Monitoring database instance connections ..................... 11
<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benefiting from centralized database access control</td>
<td>11</td>
</tr>
<tr>
<td>Updating VLDB properties for ODBC connections</td>
<td>12</td>
</tr>
<tr>
<td>Managing Intelligence Server</td>
<td>14</td>
</tr>
<tr>
<td>What happens when Intelligence Server starts?</td>
<td>15</td>
</tr>
<tr>
<td>What happens when Intelligence Server stops?</td>
<td>16</td>
</tr>
<tr>
<td>Running Intelligence Server as an application or a service</td>
<td>16</td>
</tr>
<tr>
<td>Managing MicroStrategy services from the command line</td>
<td>24</td>
</tr>
<tr>
<td>Managing and monitoring projects</td>
<td>28</td>
</tr>
<tr>
<td>Managing project status, configuration, or security:</td>
<td></td>
</tr>
<tr>
<td>Project view</td>
<td>29</td>
</tr>
<tr>
<td>Managing clustered Intelligence Servers: Cluster view</td>
<td>30</td>
</tr>
<tr>
<td>Setting the status of a project</td>
<td>31</td>
</tr>
<tr>
<td>Project and data warehouse maintenance example</td>
<td>37</td>
</tr>
<tr>
<td>Processing jobs</td>
<td>38</td>
</tr>
<tr>
<td>Intelligence Server job processing (common to all jobs)</td>
<td>38</td>
</tr>
<tr>
<td>Processing report execution</td>
<td>40</td>
</tr>
<tr>
<td>Processing object browsing</td>
<td>43</td>
</tr>
<tr>
<td>Processing element browsing</td>
<td>45</td>
</tr>
<tr>
<td>Processing Report Services document execution</td>
<td>47</td>
</tr>
<tr>
<td>Processing HTML document execution</td>
<td>49</td>
</tr>
<tr>
<td>Client-specific job processing</td>
<td>51</td>
</tr>
<tr>
<td>Monitoring currently executing jobs</td>
<td>55</td>
</tr>
<tr>
<td>Using automated installation techniques</td>
<td>56</td>
</tr>
<tr>
<td>Using a Response file to install the product</td>
<td>56</td>
</tr>
<tr>
<td>Using a Response file to configure the product</td>
<td>57</td>
</tr>
<tr>
<td>Running a silent installation</td>
<td>57</td>
</tr>
<tr>
<td>Security checklist before deploying the system</td>
<td>58</td>
</tr>
<tr>
<td>2. Setting Up User Security</td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>61</td>
</tr>
<tr>
<td>The MicroStrategy user model</td>
<td>62</td>
</tr>
<tr>
<td>About MicroStrategy users</td>
<td>62</td>
</tr>
<tr>
<td>About MicroStrategy user groups</td>
<td>63</td>
</tr>
<tr>
<td>Privileges</td>
<td>65</td>
</tr>
<tr>
<td>Permissions</td>
<td>66</td>
</tr>
<tr>
<td>Creating, importing, and deleting users and groups</td>
<td>67</td>
</tr>
<tr>
<td>Monitoring users’ connections to projects</td>
<td>68</td>
</tr>
<tr>
<td>Controlling access to application functionality</td>
<td>69</td>
</tr>
<tr>
<td>Controlling access to objects: Permissions</td>
<td>69</td>
</tr>
<tr>
<td>Controlling access to functionality: Privileges</td>
<td>79</td>
</tr>
<tr>
<td>Defining sets of privileges: Security roles</td>
<td>82</td>
</tr>
</tbody>
</table>
Controlling access to data ........................................................... 88
  Controlling access to the database: Connection mappings ... 88
  Linking database users and MicroStrategy users:
    Passthrough execution .......................................................... 93
  Restricting access to data: Security filters .................................. 95
  Controlling access to data at the database (RDBMS) level .......................................................... 109
Merging users or groups............................................................ 113
  How users and groups are merged...................................... 113
  Running the User Merge Wizard........................................ 116

3. Identifying Users: Authentication

Introduction .............................................................................. 117
  Modes of authentication............................................................. 117
    Configuring the authentication mode for a project source ... 118
    Importing users from different authentication systems ....... 119
    Sharing user accounts between users..................................... 120
  Implementing standard authentication.................................. 120
    Steps to implement standard authentication........................ 121
  Implementing anonymous authentication ............................... 122
  Implementing database warehouse authentication .................... 123
    Database warehouse authentication information flow .... 124
    Steps to implement database warehouse authentication .... 125
    Database warehouse and metadata (6.x) authentication .... 126
  Implementing LDAP authentication ........................................... 126
    LDAP information flow ........................................................ 127
    Implementing LDAP authentication in MicroStrategy......... 128
    Defining LDAP search filters to verify and import users and groups at login ........................................ 135
    Authenticating LDAP users and groups in MicroStrategy.... 139
    Synchronizing imported LDAP users and groups .......... 154
    Managing LDAP authentication in MicroStrategy .......... 159
  Enabling single sign-on authentication ...................................... 169
    Implementing Windows NT authentication.......................... 169
    Enabling integrated authentication ........................................ 176
    Enabling Single Sign-on to Web, Mobile, and Office with third-party authentication .................................. 199
  Authentication examples........................................................ 217
    Security views: Windows authentication and linked
    warehouse login ................................................................. 217
    Connection maps: Standard authentication, connection
maps, and partitioned fact tables ................................................. 218

4. Managing Your Licenses

Introduction .................................................................................. 221
Managing and verifying your licenses ........................................ 222
Named User licenses ................................................................. 222
CPU licenses ................................................................. 224
Effects of being out of compliance with your licenses ............ 225
Auditing and updating licenses ................................................. 225
Using License Manager .......................................................... 226
Auditing your system for the proper licenses ......................... 228
Updating your license ............................................................ 229
Updating CPU affinity ........................................................... 230
CPU affinity for Intelligence Server on Windows ................. 230
CPU affinity for Intelligence Server on UNIX/Linux .......... 231
CPU affinity for MicroStrategy Web ....................................... 235

5. Managing Your Projects

Introduction .............................................................................. 239
The project life cycle ................................................................. 240
Recommended scenario: Development, test, and production ........ 241
Real-life scenario: New version from a project developer..... 243
Implementing the recommended life cycle ......................... 244
Duplicating a project ............................................................... 246
What objects are duplicated with a project? ......................... 247
Duplicating projects in multiple languages ......................... 248
The Project Duplication Wizard ............................................ 249
Migrating a project to a new database platform ................. 251
Updating projects with new objects ........................................ 253
Comparing Project Merge to Object Manager ....................... 254
Locking projects ................................................................. 255
Copying objects between projects: Object Manager .......... 256
Prerequisites for copying objects between projects .......... 256
Project locking with Object Manager ..................................... 257
Copying objects ............................................................... 258
What happens when you copy or move an object ............... 262
Resolving conflicts when copying objects ......................... 269
Copying objects in a batch: Update packages ....................... 276
Merging projects to synchronize objects ......................... 296
What happens when you merge projects ......................... 297
Merging projects with the Project Merge Wizard ....................... 298
Resolving conflicts when merging projects ................................ 303
Comparing and tracking projects ............................................. 304
Comparing objects between two projects ................................. 305
Tracking your projects with the Search Export feature .............. 307
Listing unused objects in a project ........................................ 308
Deleting unused schema objects: managed objects ................. 309
Deleting managed objects one-by-one .................................... 309
Deleting all unused managed objects ..................................... 310

6. Monitoring System Usage

Introduction ............................................................................... 313
MicroStrategy system monitors .............................................. 313
Monitoring system activity: Change journaling ......................... 315
Enabling change journaling .................................................... 315
Viewing the change journal entries ....................................... 318
Increasing the number of change journal entries to view or export .......................................................... 319
Searching the change journal for relevant entries ................. 319
Exporting the change journal ................................................. 321
Purging the change journal ..................................................... 322
Monitoring system usage: Intelligence Server statistics ............ 323
Overview of Intelligence Server statistics ............................ 323
Best practices for recording Intelligence Server statistics ......... 328
Configuring Intelligence Server to log statistics .................... 330
Improving database performance by purging statistics data ............ 336
Additional monitoring tools .................................................. 338
Diagnostics and Performance Logging Tool ......................... 338
Integrity Manager ............................................................... 339
Enterprise Manager .............................................................. 339
Health Center ..................................................................... 339
Windows Performance Monitor ............................................ 340

7. Tuning your System for Best Performance

Introduction .............................................................................. 341
Tuning overview and best practices ...................................... 342
Defining the system requirements ....................................... 343
Configuring the system design ............................................. 343
Best practices for tuning your system .................................... 346
Designing system architecture .............................................. 347
How the data warehouse can affect performance ................. 347
Contents

How the network can affect performance ........................................ 348
How clustering can affect performance ........................................ 351

Managing system resources .......................................................... 352
Managing system memory and resources: Windows
Performance Monitor .................................................................... 353
Processor type, speed, and number of processors ....................... 353
Physical disk ............................................................................... 354
Memory ..................................................................................... 355

Governing Intelligence Server memory use with
Memory Contract Manager .......................................................... 360

Governing memory for requests from MicroStrategy
Web products ............................................................................. 370

Managing user sessions .............................................................. 371
Managing user sessions
Governing active users .................................................................. 372
Governing user resources ........................................................... 374
Governing user profiles ............................................................. 377

Governing requests ..................................................................... 379
Excluding document datasets from the job limits ......................... 380
Limiting the total number of jobs .................................................. 380
Limiting the number of jobs per project ....................................... 381
Limiting the number of jobs per user session and per user account ............................................................................. 381
Limiting the number of executing jobs per user and project ............ 382
Limiting the maximum report execution time .................................. 382
Limiting a report's SQL per pass .................................................. 385

Managing job execution .............................................................. 386
Managing database connection threads ........................................ 386
Prioritizing jobs .......................................................................... 390
Results processing ....................................................................... 394

Governing results delivery ........................................................... 399
Limiting the information displayed at one time .......................... 400
Limiting the number of XML cells .............................................. 401
Limiting export sizes ................................................................. 402
Limiting the memory consumption for file generation ................ 403
Limiting the total number of XML drill paths ................................. 404

Designing reports ....................................................................... 404
Analytic complexity .................................................................... 405
Subtotals .................................................................................... 405
Page-by feature .......................................................................... 406
Prompt complexity ...................................................................... 406
Report Services documents ......................................................... 406
8. Clustering Multiple MicroStrategy Servers

Introduction ........................................................................................................... 421
Overview of clustering .................................................................................... 422
Benefits of clustering ....................................................................................... 422
The clustered architecture ................................................................................ 425
Synchronizing cached information across nodes in a cluster ....................... 426
Prerequisites for clustering Intelligence Servers ........................................... 431
Clustering Intelligence Servers ...................................................................... 435
Configuring caches in a cluster ....................................................................... 435
Joining the nodes in a cluster .......................................................................... 441
Verifying the clustered system is working .................................................... 442
Distributing projects across nodes in a cluster ............................................. 443
Managing your clustered system ..................................................................... 445
Managing your projects across nodes of a cluster ......................................... 446
Project failover and latency ............................................................................ 448
Shutting down a node ....................................................................................... 451
Maintaining result caches in a clustered environment ................................ 452
Connecting MicroStrategy Web to a cluster ................................................. 453
Clustering and firewalls ................................................................................... 454
Exporting to PDF or Excel .............................................................................. 454
Node failure ......................................................................................................... 454

9. Improving Report and Document Response Time: Caching

Introduction ......................................................................................................... 455
Result caches ....................................................................................................... 456
Cache management best practices .................................................................... 459
Types of result caches ........................................................................................ 460
Location of result caches .................................................................................... 461
Cache matching algorithm ................................................................................ 463
Disabling result caching .................................................................................... 466
## Contents

- Monitoring result caches ...................................................... 466
- Managing result caches ....................................................... 470
- Configuring result cache settings ......................................... 476

### Saving report results: History List .............................................. 488
- Understanding History Lists ................................................. 489
- History Lists and caching ..................................................... 491
- Configuring History List data storage .................................... 492
- History List backup frequency ............................................. 496
- History Lists in a clustered environment ............................... 496
- Accessing History Lists ....................................................... 496
- Archiving History List messages .......................................... 498
- Managing History Lists ....................................................... 500

### Element caches ......................................................................... 504
- Element caching terminology .............................................. 506
- Location of element caches ................................................. 506
- Cache matching algorithm ................................................. 507
- Enabling or disabling element caching .................................. 507
- Limiting the number of elements displayed and cached at a time .......................................................... 508
- Caching algorithm ......................................................... 511
- Limiting the amount of memory available for element caches .......................................................... 512
- Limiting which attribute elements a user can see ............... 513
- Limiting element caches by database connection .............. 514
- Limiting element caches by database login ...................... 515
- Deleting all element caches ................................................. 516
- Summary table of element cache settings .......................... 516

### Object caches ............................................................................ 517
- Cache matching algorithm ................................................. 518
- Enabling or disabling object caching .................................... 518
- Limiting the amount of memory available for object caches .......................................................... 519
- Deleting object caches ........................................................ 520
- Summary table of object caching settings .......................... 521

## 10. Managing Intelligent Cubes

### Introduction .............................................................................. 523

- Managing Intelligent Cubes: Intelligent Cube Monitor ............... 524
- Viewing Intelligent Cube information and usage statistics .... 525
- Monitoring and modifying Intelligent Cube status ............... 526
- Governing Intelligent Cube memory usage, loading, and storage .......................................................... 532
11. Scheduling Jobs and Administrative Tasks

Introduction .............................................................................. 545

Best practices for scheduling jobs and administrative tasks ...... 546
Creating and managing schedules ............................................ 547
  Time-triggered schedules .................................................... 548
  Event-triggered schedules ................................................... 548
Creating schedules .............................................................. 548
Managing schedules ............................................................ 549
About events and event-triggered schedules ......................... 550
Scheduling reports and documents: Subscriptions ................. 553
  Types of subscriptions ......................................................... 554
Creating subscriptions .......................................................... 555
Managing your subscriptions ............................................... 559
Scheduling administrative tasks ................................................ 560
  Managing scheduled administration tasks ......................... 562
Scheduling deliveries to email, file, and printer using
  Distribution Services .......................................................... 564
    About Distribution Services ............................................. 564
    Administering report delivery .......................................... 565
    About the delivery process ............................................. 567
    Best practices for using Distribution Services features ...... 568
    High-level checklist to set up a report delivery system ....... 570
Creating and managing transmitters .................................... 571
Creating and managing devices .......................................... 577
Creating and managing contacts ......................................... 585
Subscription and delivery tasks for users ............................ 593

12. Automating Administrative Tasks with Command Manager

Introduction .............................................................................. 595

Using Command Manager ....................................................... 595
  Privileges required for using Command Manager ................ 596
  Creating and executing scripts ........................................... 597
  Procedures in Command Manager ....................................... 599
Executing a Command Manager script ................................ 600
  Encrypting Command Manager scripts .............................. 604
13. Verifying Reports and Documents with Integrity Manager

**Introduction** ................................................................. 611

What is an integrity test? ................................................... 612
  Types of integrity tests ................................................... 613
  Testing Intelligence Server performance .......................... 614

Best practices for using Integrity Manager .......................... 616

Creating an integrity test .................................................. 617
  Saving and loading a test ............................................... 619

Executing an integrity test ................................................ 620
  Executing a test against a remote Intelligence Server .......... 620
  Executing prompted reports with Integrity Manager .......... 621
  Executing a test under multiple MicroStrategy user accounts ................................................... 625
  Ignoring dynamic SQL when comparing SQL .................... 627
  Matching equivalent SQL strings .................................... 628
  Executing a test from the command line ........................... 628

Viewing the results of a test .............................................. 631
  Viewing and editing notes ............................................. 633
  Grouping differences in graph and PDF reports ................. 633
  Accessing the saved results of a test ............................... 634

14. Analyzing System Usage with Enterprise Manager

**Introduction** ................................................................. 639

What is Enterprise Manager? ............................................ 640

Best practices for using Enterprise Manager ....................... 641

Understanding the Enterprise Manager architecture ............ 641
  Enterprise Manager system components .......................... 642
  Enterprise Manager processes ...................................... 643

Installing and configuring Enterprise Manager .................... 645
  Best practices for installing and configuring Enterprise Manager ........................................... 646
  MicroStrategy Enterprise Manager prerequisites ................ 647
  Installing Enterprise Manager ....................................... 648
Configuring your projects to log statistics ......................... 648
Creating the Enterprise Manager metadata database .......... 649
Running the Enterprise Manager console ......................... 650
Upgrading Enterprise Manager ............................................ 659
Maintaining Enterprise Manager ............................................. 659
Data loading ........................................................................ 660
Enterprise Manager maintenance tasks ............................ 665
Improving warehouse performance by purging unused data ................................................................. 668
Reporting in Enterprise Manager ........................................... 669
Best practices for Enterprise Manager reporting ................. 671
Customizing Enterprise Manager reports to suit your needs ................................................................. 672
Dashboards ........................................................................ 673
Operations analysis ............................................................ 684
Performance analysis ......................................................... 693
Project analysis ................................................................... 697
Real-time analysis ............................................................... 705
User analysis ........................................................................ 706

15. Maintaining your MicroStrategy System with Health Center

Introduction ........................................................................... 711
Accessing the Health Center console ................................. 712
Configuring and managing a Health Center system .......... 712
Best practices for configuring a Health Center system ...... 713
The Master Health Agent ..................................................... 714
Health Agents .................................................................... 715
Adding Health Agents to the system ................................ 716
Diagnosing and fixing problems ......................................... 718
Fixing problems found by Health Center ......................... 720
Analyzing log files ............................................................... 720
Collecting and sending information for MicroStrategy
Technical Support ................................................................. 722
List of system checks .......................................................... 723
System checks for Configuration ....................................... 725
System checks for Desktop ............................................... 725
System checks for Enterprise Manager ............................ 726
System checks for Intelligence Server ............................... 727
System checks for Narrowcast Server .............................. 731
System checks for MicroStrategy Web products ................. 733
16. Troubleshooting

Introduction .............................................................................. 737
Methodology for finding trouble spots ........................................ 738
Finding trouble spots using diagnostics ....................................... 740
Configuring what is logged ..................................................... 740
Viewing and analyzing log files ............................................ 752
Analyzing a server state dump ............................................. 755
Memory depletion troubleshooting ............................................. 759
Basic memory concepts ....................................................... 760
Potential causes of memory depletion ....................................... 761
Authentication troubleshooting .............................................. 765
Troubleshooting Windows authentication in
MicroStrategy Web .............................................................. 766
Failure to log in to server (three-tier) project source ............ 767
Troubleshooting LDAP authentication .................................... 767
Fixing inconsistencies in the metadata ...................................... 773
Object dependencies troubleshooting ...................................... 776
Date/time functions troubleshooting ....................................... 776
Performance troubleshooting ..................................................... 777
Project performance ......................................................... 777
Schema update performance ............................................... 778
Cache Monitor and Intelligent Cube Monitor performance ............. 778
Troubleshooting data imported from a file ................................. 779
Subscription and report results troubleshooting ................. 780
Drilled-to report returns no data or incorrect data .............. 780
Graph report results ............................................................. 780
Number of report result rows .............................................. 780
Subscription with a prompt is not sent and disappears ........ 781
Contact does not appear in list of contacts when
subscribing ........................................................................... 781
SQL/MDX string length ........................................................ 782
Freeform SQL report error ................................................... 783
Internationalization troubleshooting ...................................... 784
Object displays translations in a different language
than project or object default language ................................ 784
User’s preferred language defaults to project default
language ................................................................................ 784
Custom names for system objects are not retained after
upgrade ............................................................................... 785
Failure to start MicroStrategy Intelligence Server .............. 785
Logon failure ....................................................................... 785
## Failure to activate Intelligence Server

- 786

## Clustered environments troubleshooting

- 787
  - Problems in a clustered environment
    - 787
  - Caches in a clustered environment
    - 788
  - Node synchronization in a clustered environment
    - 789

## Statistics logging troubleshooting

- 790
  - Statistics are not being logged
    - 790
  - Statistics on subscriptions and deliveries
    - 793

## Additional resources for troubleshooting

- 794
  - MicroStrategy readme and release notes
    - 794
  - MicroStrategy Knowledge Base
    - 795
  - Customer Forums
    - 796
  - Technical Support
    - 797

## Administering MicroStrategy Web and Mobile

### Introduction

- 799

### Assigning privileges for MicroStrategy Web

- 800

### Using the MicroStrategy Web Administrator page

- 802
  - Controlling access to the Administrator page
    - 802

### Defining project defaults

- 804
  - Loading and applying default values
    - 804

### Using additional security features for MicroStrategy Web and Mobile

- 805
  - Using firewalls
    - 806
  - Working with SSL (HTTPS)
    - 809
  - Using cookies
    - 810
  - Using encryption
    - 812
  - Applying file-level security
    - 813
  - Sample MicroStrategy system
    - 814

### Integrating Narrowcast Server with MicroStrategy Web products

- 815

### Enabling users to install MicroStrategy Office from Web

- 816

### FAQs for configuring and tuning MicroStrategy Web products

- 817

### Glossary

- 821

### Index

- 835
BOOK OVERVIEW AND ADDITIONAL RESOURCES

Description of 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:

•  Chapter 1, 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 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 3, 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 4, 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 5, 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 6, 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 7, 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 8, Clustering Multiple MicroStrategy Servers**

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 9, 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 Web and Desktop, and how to administer it.

• **Chapter 10, 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 11, 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 12, 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 13, Verifying Reports and Documents with Integrity Manager**
  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 14, Analyzing System Usage with Enterprise Manager**
  Enterprise Manager provides insights about governing and tuning all areas of your MicroStrategy environment. It provides a historical overview of system usage. This chapter introduces you to Enterprise Manager, provides best practices, and shows you how to install and configure Enterprise Manager. It also describes how to analyze statistics and generate usage reports with Enterprise Manager.

- **Chapter 15, 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 16, 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.

- **Chapter 17, 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 Server, Web Server Universal, and Mobile Server.
About this book

This book is divided into chapters that begin with a brief overview of the chapter’s content.

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

Dates in the MicroStrategy Tutorial project are updated to reflect the current year. The sample documents and images in this guide, as well as the procedures, were created with dates that may no longer be available in the Tutorial project. Replace them with the first year of data in your Tutorial project.

Additional formats

This book is also available as an electronic publication in the Apple iBookstore, and can be read on an iPhone or iPad with the iBooks app installed. To download this book, search for the book’s title in the iBookstore search bar, or scan the QR code below using your device's camera.

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, metadata, and project. Information about the MicroStrategy Tutorial can be found in the MicroStrategy Basic Reporting Guide.

Detailed examples of advanced reporting functionality can be found in the MicroStrategy Advanced Reporting Guide.
Other examples in this book use the Analytics Modules, which include a set of precreated sample reports, each from a different business area. Sample reports present data for analysis in such business areas as financial reporting, human resources, and customer analysis.

What’s new in this guide

**MicroStrategy 9.2.1**

- The following information is now included in the new *Supplemental Reference for System Administration*:
  - VLDB properties
  - Metadata and data warehouse internationalization
  - Privileges
  - MicroStrategy Web cookies
  - Command Manager Runtime statement syntax guide
  - Data dictionary for Intelligence Server statistics tables

- The following information is now included in the System Administration Guide:
  - *Chapter 12, Automating Administrative Tasks with Command Manager*
  - *Chapter 13, Verifying Reports and Documents with Integrity Manager*
  - *Chapter 14, Analyzing System Usage with Enterprise Manager*
  - *Chapter 15, Maintaining your MicroStrategy System with Health Center*

**MicroStrategy 9.2**

- Personalize email and file subscriptions with macros for values such as subscription name and report or document name (see *Personalizing*)
email and file subscriptions, page 558 and Personalizing file locations, page 580).

- Deliver Distribution Services file subscriptions from a UNIX/Linux Intelligence Server to a Windows file location using Sharity™ (see Delivering files from a UNIX Intelligence Server to a Windows file location, page 581).

- Control access to Intelligent Cubes with the new permission groupings Consume, Add, and Collaborate (see What permissions can be granted for an object?, page 72).

- Import objects from update packages using two new conflict resolution rules: Force Replace and Delete (see About update packages, page 277).

- In Object Manager, create update packages directly from the Conflict Resolution dialog box, containing all objects listed in the Conflict
Resolution dialog box (see To create a project update package, page 279 and To create a configuration update package, page 282).


- Keep the size of Intelligent Cubes in check by limiting the size of indexes. See Defining limits for Intelligent Cube indexes, page 538.

- Best practices information for configuring a Health Center network (see Best practices for configuring a Health Center system, page 713).

- New Health Center system checks for Intelligence Server and MicroStrategy Web products (see List of system checks, page 723).

- Best practices information for using Integrity Manager (see Best practices for using Integrity Manager, page 616).

- Execute integrity tests under multiple user accounts (see Executing a test under multiple MicroStrategy user accounts, page 625).

- Answer prompts in integrity tests using personal answers other than the default personal answer (see Using non-default personal answers in prompts, page 622).

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
Resources

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 xxxiii*. Most of these manuals are also available printed in a bound, soft cover format. To purchase printed manuals, contact your MicroStrategy account executive or email documentationfeedback@microstrategy.com.

- **Help**: In general, MicroStrategy help provides:
  - Detailed steps to perform procedures
  - Descriptions of each option on every software screen

  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. For the most up-to-date translations, refer to the MicroStrategy Knowledge Base.

**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.

• **Evaluate MicroStrategy for Linux Guide: In a Windows or Linux Environment with the MicroStrategy Evaluation Edition Virtual Appliance**

Evaluate MicroStrategy for Linux, in a Microsoft Windows or Linux environment, with the MicroStrategy Evaluation Edition Virtual Appliance. This guide provides all details to download, activate, and evaluate MicroStrategy software running in a Linux environment.

• **MicroStrategy Reporting Suite: Quick Start Guide**

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

• **MicroStrategy Mobile Suite: Quick Start Guide**

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

**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.

• **MicroStrategy Basic Reporting Guide**

Instructions to get started with MicroStrategy Desktop 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.


Instructions for a business analyst to execute and analyze a document in MicroStrategy Desktop and MicroStrategy Web, building on basic concepts about projects and reports presented in the *MicroStrategy Basic Reporting Guide*.

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


• **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, Word, and Outlook, 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**

Information and instructions to install and configure MicroStrategy Mobile, as well as instructions for a designer working in MicroStrategy Desktop 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.

• **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 sold as 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 using the MicroStrategy Product Manuals page, as described in Accessing manuals and other documentation sources, page xxxiii.

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 ESRI mapping services.

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

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 your MicroStrategy disk or 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 installed manuals and other documentation sources on Windows, page xxxiii
- To access installed manuals and other documentation sources on UNIX and Linux, page xxxiii

To access installed manuals and other documentation sources on Windows

1. From the Windows Start menu, choose Programs (or All Programs), MicroStrategy, 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 installed manuals and other documentation sources 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 Documentation 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>• 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</td>
</tr>
<tr>
<td></td>
<td>• Text to be entered by the user</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Click <strong>Select Warehouse.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Type <strong>cmdmgr -f scriptfile.scp</strong> and press <strong>Enter.</strong></td>
</tr>
<tr>
<td><strong>italic</strong></td>
<td>• New terms defined within the text and in the glossary</td>
</tr>
<tr>
<td></td>
<td>• Names of other product manuals</td>
</tr>
<tr>
<td></td>
<td>• When part of a command syntax, indicates variable information to be replaced by the user</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> The <strong>aggregation level</strong> is the level of calculation for the metric.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Type <strong>copy c:\filename d:\foldername\filename</strong></td>
</tr>
</tbody>
</table>
### Documentation standards

MicroStrategy 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 guide; some languages have rules that supersede the table below.

<table>
<thead>
<tr>
<th>Type</th>
<th>Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Courier font</strong></td>
<td>• Calculations</td>
</tr>
<tr>
<td></td>
<td>• Code samples</td>
</tr>
<tr>
<td></td>
<td>• Registry keys</td>
</tr>
<tr>
<td></td>
<td>• Path and file names</td>
</tr>
<tr>
<td></td>
<td>• URLs</td>
</tr>
<tr>
<td></td>
<td>• Messages displayed in the screen</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> ( \text{Sum(revenue)} / \text{number of months} ).</td>
</tr>
<tr>
<td>+</td>
<td>A keyboard command that calls for the use of more than one key (for example,</td>
</tr>
<tr>
<td></td>
<td>\text{SHIFT+F1})</td>
</tr>
<tr>
<td>!</td>
<td>A note icon indicates helpful information for specific situations.</td>
</tr>
<tr>
<td>!</td>
<td>A warning icon alerts you to important information such as potential security</td>
</tr>
<tr>
<td></td>
<td>risks; these should be read before continuing.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>• Button names, check boxes, options, lists, and menus that are the focus of</td>
</tr>
<tr>
<td></td>
<td>actions or part of a list of such GUI elements and their definitions</td>
</tr>
<tr>
<td></td>
<td>• Text to be entered by the user</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Click <strong>Select Warehouse</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Type <strong>cmdmgr -f scriptfile.scp</strong> and press <strong>Enter</strong>.</td>
</tr>
<tr>
<td><strong>italic</strong></td>
<td>• New terms defined within the text and in the glossary</td>
</tr>
<tr>
<td></td>
<td>• Names of other product manuals</td>
</tr>
<tr>
<td></td>
<td>• When part of a command syntax, indicates variable information to be</td>
</tr>
<tr>
<td></td>
<td>replaced by the user</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> The <strong>aggregation level</strong> is the level of calculation for the</td>
</tr>
<tr>
<td></td>
<td>metric.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Type <strong>copy c:\filename d:\foldername\filename</strong></td>
</tr>
<tr>
<td><strong>Courier font</strong></td>
<td>• Calculations</td>
</tr>
<tr>
<td></td>
<td>• Code samples</td>
</tr>
<tr>
<td></td>
<td>• Registry keys</td>
</tr>
<tr>
<td></td>
<td>• Path and file names</td>
</tr>
<tr>
<td></td>
<td>• URLs</td>
</tr>
<tr>
<td></td>
<td>• Messages displayed in the screen</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> ( \text{Sum(revenue)} / \text{number of months} ).</td>
</tr>
</tbody>
</table>
## 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](http://www.microstrategy.com/Education).

## Consulting

MicroStrategy Consulting Services provides proven methods for delivering leading-edge technology solutions. Offerings include complex security architecture designs, performance and tuning, project and testing strategies and recommendations, strategic planning, and more. For a detailed description of consulting offerings, visit [http://www.microstrategy.com/Consulting](http://www.microstrategy.com/Consulting).

## International support

MicroStrategy supports several locales. Support for a locale typically includes native database and operating system support, support for date formats,
numeric formats, currency symbols, and availability of translated interfaces and certain documentation.

MicroStrategy is certified in homogeneous configurations (where all the components lie in the same locale) in the following languages—English (US), French, German, Italian, Japanese, Korean, Portuguese (Brazilian), Spanish, Chinese (Simplified), Chinese (Traditional), Danish, and Swedish. A translated user interface is available in each of the above languages. For information on specific languages supported by individual MicroStrategy system components, see the MicroStrategy readme.

MicroStrategy also provides limited support for heterogeneous configurations (where some of the components may lie in different locales). Please contact MicroStrategy Technical Support for more details.

A translated user interface is available in each of the above languages.

**Technical Support**

If you have questions about a specific MicroStrategy product, you should:

1. Consult the product guides, Help, and readme files. Locations to access each are described above.

2. Consult the MicroStrategy Knowledge Base online at [https://resource.microstrategy.com/support](https://resource.microstrategy.com/support).

   A technical administrator in your organization may be able to help you resolve your issues immediately.

3. If the resources listed in the steps above do not provide a solution, contact MicroStrategy Technical Support directly. To ensure the most productive relationship with MicroStrategy Technical Support, review the Policies and Procedures document in your language, posted at [http://www.microstrategy.com/Support/Policies](http://www.microstrategy.com/Support/Policies). Refer to the terms of your purchase agreement to determine the type of support available to you.

MicroStrategy Technical Support can be contacted by your company’s Support Liaison. A Support Liaison is a person whom your company has designated as a point-of-contact with MicroStrategy’s support personnel. All customer inquiries and case communications must come through these named individuals. Your company may designate two employees to serve as
their Support Liaisons, and can request to change their Support Liaisons two
times per year with prior written notice to MicroStrategy Technical Support.

It is recommended that you designate Support Liaisons who have
MicroStrategy Administrator privileges. This can eliminate security conflicts
and improve case resolution time. When troubleshooting and researching
issues, MicroStrategy Technical Support personnel may make
recommendations that require administrative privileges within
MicroStrategy, or that assume that the designated Support Liaison has a
security level that permits them to fully manipulate the MicroStrategy
projects and has access to potentially sensitive project data such as security
filter definitions.

Ensure issues are resolved quickly

Before logging a case with MicroStrategy Technical Support, the Support
Liaison may follow the steps below to ensure that issues are resolved quickly:

1 Verify that the issue is with MicroStrategy software and not a third party
   software.

2 Verify that the system is using a currently supported version of
   MicroStrategy software by checking the Product Support Expiration

3 Attempt to reproduce the issue and determine whether it occurs
   consistently.

4 Minimize the complexity of the system or project object definition to
   isolate the cause.

5 Determine whether the issue occurs on a local machine or on multiple
   machines in the customer environment.

6 Discuss the issue with other users by posting a question about the issue
   on the MicroStrategy Customer Forum at
   https://resource.microstrategy.com/forum/.

The following table shows where, when, and how to contact MicroStrategy
Technical Support. If your Support Liaison is unable to reach MicroStrategy
Technical Support by phone during the hours of operation, they can leave a
voicemail message, send email or fax, or log a case using the Online Support
Interface. The individual Technical Support Centers are closed on certain public holidays.

<table>
<thead>
<tr>
<th>Region</th>
<th>Email</th>
<th>Web</th>
<th>Phone and Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td><a href="mailto:support@microstrategy.com">support@microstrategy.com</a></td>
<td><a href="https://resource.microstrategy.com/support">https://resource.microstrategy.com/support</a></td>
<td>(703) 848–8700, Hours: 9:00 A.M.–7:00 P.M. Eastern Time, Monday–Friday except holidays</td>
</tr>
<tr>
<td>EMEA: Europe</td>
<td><a href="mailto:eurosupp@microstrategy.com">eurosupp@microstrategy.com</a></td>
<td><a href="https://resource.microstrategy.com/support">https://resource.microstrategy.com/support</a></td>
<td>+44 (0) 208 711 2525, The European Technical Support Centre is closed on national public holidays in each country.</td>
</tr>
<tr>
<td>The Middle East</td>
<td></td>
<td></td>
<td>Phone:  • Belgium: + 32 2792 0436  • France: +33 17 099 4737  • Germany: +49 22 16501 0609  • Ireland: +353 1436 0916  • Italy: +39 023626 9668  • Poland: +48 22 321 8680  • Scandinavia &amp; Finland: +46 8505 20421  • Spain: +34 91788 9852  • The Netherlands: +31 20 794 8425  • UK: +44 (0) 208 080 2182  • International distributors: +44 (0) 208 080 2183</td>
</tr>
<tr>
<td>Africa</td>
<td></td>
<td></td>
<td>Hours:  • United Kingdom: 9:00 A.M.–6:00 P.M. GMT, Monday–Friday except holidays  • EMEA (except UK): 9:00 A.M.–6:00 P.M. CET, Monday–Friday except holidays</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td><a href="mailto:apsupport@microstrategy.com">apsupport@microstrategy.com</a></td>
<td><a href="https://resource.microstrategy.com/support">https://resource.microstrategy.com/support</a></td>
<td>+61 2 9333 6499, Phone:  • Australia: +61 2 9333 6499  • Korea: +82 2 560 6565 Fax: +82 2 560 6555  • Japan: +81 3 3511 6720 Fax: +81 3 3511 6740  • Singapore: +65 6303 8969 Fax: +65 6303 8999  • Asia Pacific (except Australia, Japan, Korea, and Singapore): +86 571 8526 8067 Fax: +86 571 8848 0977</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hours:  • Japan and Korea: 9:00 A.M.–6:00 P.M. JST (Tokyo), Monday–Friday except holidays  • Asia Pacific (except Japan and Korea): 7 A.M.–6 P.M. (Singapore) Monday–Friday except holidays</td>
</tr>
<tr>
<td>Latin America</td>
<td><a href="mailto:latamsupport@microstrategy.com">latamsupport@microstrategy.com</a></td>
<td><a href="https://resource.microstrategy.com/support">https://resource.microstrategy.com/support</a></td>
<td>+54 11 5222 9360 Fax: +54 11 5222 9355, Phone:  • LATAM (except Brazil and Argentina): +54 11 5222 9360 Fax: +54 11 5222 9355  • Argentina: 0 800 444 MSTR Fax: +54 11 5222 9355  • Brazil: +55 11 3054 1010 Fax: +55 11 3044 4088</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hours:  • Latin America (except Brazil): 9:00 A.M.–7:00 P.M. (Buenos Aires), Monday–Friday except holidays  • Brazil: 9 A.M. - 6 P.M. (São Paulo), Monday–Friday except holidays</td>
</tr>
</tbody>
</table>
Support Liaisons should contact the Technical Support Center from which they obtained their MicroStrategy software licenses or the Technical Support Center to which they have been designated.

**Required information when calling**

When contacting MicroStrategy Technical Support, please provide the following information:

- **Personal information:**
  - Name (first and last)
  - Company and customer site (if different from company)
  - Contact information (phone and fax numbers, e-mail addresses)

- **Case details:**
  - Configuration information, including MicroStrategy software product(s) and versions
- Full description of the case including symptoms, error messages(s), and steps taken to troubleshoot the case thus far

- **Business/system impact**

If this is the Support Liaison’s first call, they should also be prepared to provide the following:

- Street address
- Phone number
- Fax number
- Email address

To help the Technical Support representative resolve the problem promptly and effectively, be prepared to provide the following additional information:

- **Case number:** Please keep a record of the number assigned to each case logged with MicroStrategy Technical Support, and be ready to provide it when inquiring about an existing case

- **Software version and product registration numbers of the MicroStrategy software products you are using**

- **Case description:**
  - What causes the condition to occur?
  - Does the condition occur sporadically or each time a certain action is performed?
  - Does the condition occur on all machines or just on one?
  - When did the condition first occur?
  - What events took place immediately prior to the first occurrence of the condition (for example, a major database load, a database move, or a software upgrade)?
  - If there was an error message, what was its exact wording?
  - What steps have you taken to isolate and resolve the issue? What were the results?

- **System configuration (the information needed depends on the nature of the problem; not all items listed below may be necessary):**
  - Computer hardware specifications (processor speed, RAM, disk space, and so on)
- Network protocol used
- ODBC driver manufacturer and version
- Database gateway software version
- (For MicroStrategy Web-related problems) browser manufacturer and version
- (For MicroStrategy Web-related problems) Web server manufacturer and version

If the issue requires additional investigation or testing, the Support Liaison and the MicroStrategy Technical Support representative should agree on certain action items to be performed. The Support Liaison should perform any agreed-upon actions before contacting MicroStrategy Technical Support again regarding the issue. If the Technical Support representative is responsible for an action item, the Support Liaison may call MicroStrategy Technical Support at any time to inquire about the status of the issue.

**Feedback**

Please send any comments or suggestions about user documentation for MicroStrategy products to:

documentationfeedback@microstrategy.com

Send suggestions for product enhancements to:

support@microstrategy.com

When you provide feedback to us, please include the name and version of the products you are currently using. Your feedback is important to us as we prepare for future releases.
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 discussions in this chapter include:

- Best practices for MicroStrategy system administration, page 2
- Understanding the MicroStrategy architecture, page 3
- Communicating with databases, page 8
- Managing Intelligence Server, page 14
- Managing and monitoring projects, page 28
- Processing jobs, page 38
- Using automated installation techniques, page 56
- Security checklist before deploying the system, page 58
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 240.

• 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 pre-defined administrative security roles for this purpose. For more information about these security roles, see The role-based administration model, page 86. 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 7, 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 Chapter 14, Analyzing System Usage with Enterprise Manager.

• 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 8, 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 9, 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 add-on to Intelligence Server. For an overview of Intelligent Cubes, see Chapter 10, 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 11, Scheduling Jobs and Administrative Tasks.

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, page 4 and Indexing your data: MicroStrategy metadata, page 4.

- 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, page 6*.

If users on MicroStrategy Desktop 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, page 7*.

- The third tier in this system is the MicroStrategy Web Server, which delivers the reports to the MicroStrategy Web client. For an introduction to MicroStrategy Web Server, see *Chapter 17, Administering MicroStrategy Web and Mobile*.

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

In a three-tier system, MicroStrategy Desktop is the last tier.

For a more complete discussion of BI 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 of 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 (on which machine it is located 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 of the metadata information is encrypted and stored in database tables defined by MicroStrategy.

For more information about running the MicroStrategy Configuration Wizard, see the *MicroStrategy 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 particular 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 warehouse schema. Therefore, it is not actually stored in a location in the metadata, but it is implicitly present within 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 Desktop. The *MicroStrategy 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 MicroStrategy Desktop or Command Manager. The
MicroStrategy 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 MicroStrategy Desktop (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 MicroStrategy 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 Desktop.

Intelligence Server is the heart of the MicroStrategy system. It executes reports stored in the metadata against the data warehouse, and passes the results of those reports on to the users. For detailed information about Intelligence Server, including how to start and stop it, see Managing Intelligence Server, page 14.

A server definition is an instance of Intelligence Server and all of its configuration settings. Multiple server definitions can be stored in the metadata, but only one can be run at a time on a given 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 8, 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:

- Metadata connectivity information, such as:
  - Metadata DSN
  - Metadata ID and encrypted password
- MicroStrategy administrator user name
• MicroStrategy Intelligence Server configuration settings – set in MicroStrategy Desktop

Tying it all together: projects and project sources

A MicroStrategy *project* is an object in which you define all of 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 *MicroStrategy Project Design Guide*.

You can manage your projects using the System Administration Monitor. For details, see *Managing and monitoring projects, page 28*.

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

There are two types of project sources, defined based on 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 MicroStrategy Desktop 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), which connects directly to a MicroStrategy version 6 project in read-only mode.

For more information on project sources, see the *MicroStrategy Installation and Configuration Guide*.
Communicating with databases

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

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 11*.

Connecting to the MicroStrategy metadata

MicroStrategy users need connectivity to the metadata so that they can access projects, create objects, and execute reports. MicroStrategy 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 MicroStrategy Desktop user access the metadata?
- How does a user connect to MicroStrategy Intelligence Server?
- Where is the connection information stored?

The diagram below illustrates 3-tier metadata connectivity between the MicroStrategy metadata database (tier one), Intelligence Server (tier two), and MicroStrategy Desktop (tier three).
In a server (three-tier) environment, MicroStrategy Desktop metadata connectivity is established through the project source. For steps to create a project source, see the *MicroStrategy Installation and Configuration Guide*. The MicroStrategy Desktop connection information is stored in the MicroStrategy Desktop 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, and can be accomplished by performing the following procedures in MicroStrategy Desktop:

- Creating a database instance: A MicroStrategy object created in Desktop 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.

- 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 *MicroStrategy 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, MicroStrategy 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 might 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.

To view the current database connections

1. In Desktop, 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 MicroStrategy 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—Since only Intelligence Server connects to databases, it can make sure that no single database becomes overloaded with user requests. This is particularly 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—Since 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 based on 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.

- 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 MicroStrategy 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 of 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 based on 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 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 since 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 Help. (From within Desktop, press F1.)

The readme file for each MicroStrategy release 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 7, 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 definition is saved in the machine’s registry. Intelligence Server reads this information when it starts 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 *MicroStrategy 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 subject and steps to load MDX cube schemas when Intelligence Server starts, see the *Configuring and Connecting Intelligence Server* chapter of the *MicroStrategy Installation and Configuration Guide*.

In the event of a system or power failure, 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 will 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 MicroStrategy Desktop, 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 17.*

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 23.*
Registering and unregistering Intelligence Server as a UNIX service

In UNIX, when you configure Intelligence Server you need to specify whether you want to start it 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 need to 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.

There are two ways to register Intelligence Server as a service:

- 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 only be done from the command line, in ~/MicroStrategy/bin. The syntax to unregister the service is:

  mstrctl -s IntelligenceServer us

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, there are several reasons why you may need to stop and restart it:

- 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 MicroStrategy 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 *MicroStrategy Service Manager*, page 18.

- If you are already using MicroStrategy Desktop, you may need to start and stop Intelligence Server from within Desktop. For instructions, see *MicroStrategy Desktop*, page 21.

- You can start and stop Intelligence Server as part of a Command Manager script. For details, see *MicroStrategy Command Manager*, page 21.

- Finally, you can start and stop Intelligence Server from the command line using MicroStrategy Server Control Utility. For instructions, see *Command line*, page 21.

**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 69. For a list of the permission groupings for server definition objects, see *Permissions for server governing and configuration*, page 74.

- 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.

**MicroStrategy Service Manager**

MicroStrategy Service Manager is a management tool installed with MicroStrategy Intelligence Server that enables you to start and stop MicroStrategy Intelligence Server and choose a startup option. For detailed 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

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

If the icon is not present in the system tray, then from the Windows Start menu, point to Programs, then MicroStrategy, then Tools, then choose 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 directory specified as the home directory during MicroStrategy installation (the default is ~/MicroStrategy), then browse to /bin.

2 Type ./mstrsvcmsg and press ENTER. The MicroStrategy Service Manager dialog box opens.
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 MicroStrategy, then Tools, then choose 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 MicroStrategy, then Tools, then choose Service Manager. The MicroStrategy Service Manager dialog box opens.
2. In the Server drop-down list, select the machine on which the MicroStrategy 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**.

### MicroStrategy Desktop

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

#### To start or stop Intelligence Server using Desktop

1 In Desktop, in the Folder List, right-click the **Administration** icon.

2 Choose **Start Server** to start it or **Stop Server** to stop it.

### MicroStrategy 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 MicroStrategy Intelligence Server using MicroStrategy 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 12, 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 located in `C:\Program Files\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 24.*

Windows Services window

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

---

**To start and stop MicroStrategy 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 MicroStrategy Intelligence Server as an application. This may be necessary if you must administer Intelligence Server on the machine on which it is installed, if MicroStrategy Desktop 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 MicroStrategy 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 17.

The default path for the MicroStrategy Intelligence Server application executable is C:\Program Files\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, page 6.

Server Control Utility can also be used to start, stop, and restart other MicroStrategy services, such as the Test Listener or the Enterprise Manager Data Loader, 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 will be prompted for a password.

- **servicename** is the name of the service, such as IntelligenceServer or EMDataLoader.

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.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Get information about the Server Control Utility</strong></td>
<td></td>
</tr>
<tr>
<td>List all commands for the Server Control Utility.</td>
<td><code>-h</code> or <code>--help</code></td>
</tr>
<tr>
<td><strong>Note:</strong> This command does not require a machine name, login, or service name.</td>
<td></td>
</tr>
<tr>
<td>Display the version number of the Server Control Utility.</td>
<td><code>-V</code> or <code>--version</code></td>
</tr>
<tr>
<td><strong>Note:</strong> This command does not require a machine name, login, or service name.</td>
<td></td>
</tr>
<tr>
<td><strong>Get information about the MicroStrategy network</strong></td>
<td></td>
</tr>
<tr>
<td>List machines that the Server Control Utility can see and affect.</td>
<td><code>lm</code> or <code>list-machines</code></td>
</tr>
<tr>
<td><strong>Note:</strong> This command does not require a machine name, login, or service name.</td>
<td></td>
</tr>
<tr>
<td>List the MicroStrategy services available on a machine.</td>
<td><code>ls</code> or <code>list-servers</code></td>
</tr>
<tr>
<td><strong>Note:</strong> This command does not require a service name.</td>
<td></td>
</tr>
<tr>
<td>List the ODBC DSNs available on a machine.</td>
<td><code>lod</code> or <code>list-odbc-dsn</code></td>
</tr>
<tr>
<td><strong>Note:</strong> This command does not require a service name.</td>
<td></td>
</tr>
<tr>
<td><strong>Configure a service</strong></td>
<td></td>
</tr>
<tr>
<td>If you want to...</td>
<td>Then use this command...</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------</td>
</tr>
</tbody>
</table>
| Display the configuration information for a service, in XML format. For more information, see *Using files to store output and provide input, page 27.*  
**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 27.*  
**Note:** You can optionally specify a file to save the configuration properties to. | `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 27.*  
**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 27.*  
**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 27.*  
**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 27.*  
**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`  
`set-default-instance instancename` |
| Create a new server instance. | `ci instancename`  
`create-instance 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 provide this

<table>
<thead>
<tr>
<th>If you want to . . .</th>
<th>Then use this command . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a copy of a server instance. Specify the name for the new instance as newinstancename.</td>
<td>cpi instancename newinstancename copy-instance instancename newinstancename</td>
</tr>
<tr>
<td>Delete a server instance.</td>
<td>di instancename delete-instance instancename</td>
</tr>
<tr>
<td>Register a server instance as a service.</td>
<td>rs instancename register-service instancename</td>
</tr>
<tr>
<td>Unregister a registered server instance as a service.</td>
<td>us instancename unregister-service instancename</td>
</tr>
<tr>
<td>Display the license information for a service instance.</td>
<td>gl instancename get-license instancename</td>
</tr>
<tr>
<td>Display the status information for a server instance</td>
<td>gs instancename get-status instancename</td>
</tr>
<tr>
<td>Start or stop a server instance</td>
<td></td>
</tr>
<tr>
<td>Start a server instance as a service.</td>
<td>start --service instancename</td>
</tr>
<tr>
<td>Start a server instance as an application. For more information, see Starting Intelligence Server as an application, page 23.</td>
<td>start --interactive instancename</td>
</tr>
<tr>
<td>Stop a server instance that has been started as a service.</td>
<td>stop instancename</td>
</tr>
<tr>
<td>Pause a server instance that has been started as a service</td>
<td>pause instancename</td>
</tr>
<tr>
<td>Resume a server instance that has been started as a service and paused.</td>
<td>resume instancename</td>
</tr>
<tr>
<td>Terminate a server instance that has been started as a service.</td>
<td>term instancename terminate instancename</td>
</tr>
</tbody>
</table>
type of configuration 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 a given
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 29.
• 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 30.*

• The Scheduled Maintenance monitor, which lists all the scheduled maintenance tasks. For detailed information, see *Scheduling administrative tasks, page 560.*

**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 443.*

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 315.*
- 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 35*)

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 30.*

- 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 560.*

**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
- 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 specific machine, or idle or resume projects on a specific machine for maintenance (for details, see *Changing the status of a project, page 35*) by right-clicking a specific project on a server. For more detailed information about any of these options, see the Desktop Help, or see *Managing your projects across nodes of a cluster, page 446*.

Setting the status of a project

Each project in MicroStrategy 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 32*
- *Unloaded, page 32*
- *Request Idle, page 33*
- *Execution Idle, page 33*
- *Warehouse Execution Idle, page 34*
- *Full Idle, page 34*
- *Partial Idle, page 35*

For instructions on changing a project’s status, see *Changing the status of a project, page 35*.

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 37*.

**Loaded**

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

**Unloaded**

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

Unloading a project can be helpful when an administrator has changed some project configuration settings which 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 currently 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 their History List, then the results are available in the user’s 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, as well as jobs that are executed within 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 cancelled instead of being placed in the job queue. When the project is resumed, if the job was sent to the user’s History List then 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 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 cancelled instead of being placed in the job queue. When the project is resumed, if the job was sent to the user’s History List then 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 currently 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 currently active requests that require SQL to be submitted to the data warehouse are queued until the project is resumed. All other currently active requests are completed.

This mode allows you to stop all Intelligence Server and data warehouse processing for a project, while not cancelling 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 31.

1. In Desktop, 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 31.*

1. In Desktop, 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.

![Idle/Resume dialog box](image)

4. Select the options for the idle mode that you want to set the project to:
   - **Request Idle** (*Request Idle*): all currently executing and queued jobs finish executing, and any newly submitted jobs are rejected.
   - **Execution Idle** (*Execution Idle for All Jobs*): all currently 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 currently 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 currently executing and queued jobs are cancelled, and any newly submitted jobs are rejected.

Partial Idle (Request Idle and Execution Idle for Warehouse jobs): all currently executing and queued jobs that do not submit SQL against the data warehouse are cancelled, 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 Project2, 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 Desktop, 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 55."

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, based on factors such as the type of request, the user or user group requesting the job, the source of the job (such as Desktop, 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 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 390."

### 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 within 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 within 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 40
- Processing object browsing, page 43
- Processing element browsing, page 45
- Processing Report Services document execution, page 47
- Processing HTML document execution, page 49
- Client-specific job processing, page 51

Being familiar with this material should help you to understand and interpret statistics, Enterprise Manager reports, and other log files available within 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.

<table>
<thead>
<tr>
<th>Component</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical Engine Server</td>
<td>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.</td>
</tr>
<tr>
<td>Metadata Server</td>
<td>Controls all access to the metadata for the entire project.</td>
</tr>
<tr>
<td>Object Server</td>
<td>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.</td>
</tr>
<tr>
<td>Query Engine</td>
<td>Sends the SQL generated by the SQL Engine to the data warehouse for execution.</td>
</tr>
<tr>
<td>Report Server</td>
<td>Creates and manages all server reporting instance objects. Maintains a cache of executed reports.</td>
</tr>
<tr>
<td>Resolution Server</td>
<td>Resolves prompts for report requests. Works in conjunction with Object Server and Element Server to retrieve necessary objects and elements for a given request.</td>
</tr>
<tr>
<td>SQL Engine Server</td>
<td>Generates the SQL needed for the report.</td>
</tr>
</tbody>
</table>
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.

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 42.

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 456.

   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 cache.
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 based on 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 40*):

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 of time 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.

For information on personal Intelligent Cubes and Intelligent Cubes, see the *OLAP Services Guide*.

**Processing graph reports**

When processing graph reports, Intelligence Server performs the regular report processing (see *Processing report execution, page 40*). Depending on the connection, the following happens:

- In a three-tier connection, Intelligence Server sends the report to MicroStrategy Desktop, 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 MicroStrategy Desktop or Web, MicroStrategy
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 517*.

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

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

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

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 they are 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 *MicroStrategy 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 MicroStrategy Desktop using the Data Explorer (either in the Folder List or the Report Editor)
- Use Desktop’s Filter Editor, Custom Group Editor, or Security Filter Editor
- Use the Design Mode on 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 504*. 
The most prominent Intelligence Server components related to element browsing are listed here.

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

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

### 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 of 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 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 40*. 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 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.

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 51*
- *Processing jobs from Narrowcast Server, page 54*

For information about the processing steps performed by Intelligence Server for all jobs, see *Intelligence Server job processing (common to all jobs), page 38.*

**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 MicroStrategy Intelligence Server, which processes the job as usual (see Processing report execution, page 40).

4 Intelligence Server sends the results back to the MicroStrategy Web API via XML.

5 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 Web sends the HTML to the client’s browser, which displays the results.

**What happens when I export a report from 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 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 Web can be exported to the following formats:

- *Export to Comma Separated File (CSV) or Excel with Plain Text, page 53*
- *Export to Excel with Formatting, page 53*
- *Export to PDF, page 54*

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 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 400* 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 of 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 of 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 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 based on recipient preferences.
4 Narrowcast Server submits one report per user or one multi-page report for multiple users, depending on service definition.

5 Intelligence Server processes the report job request as usual. (See Processing report execution, page 40.) 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 currently performing. You can see jobs that are:

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

The Job Monitor displays which tasks are currently 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 Desktop, 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 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**

There are several ways to make installing of the MicroStrategy system across your enterprise easier. They are mentioned here, but more fully explained in the *MicroStrategy Installation 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 within 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 MicroStrategy 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, a MicroStrategy Intelligence Server, and multiple project sources. For more information about how to set up and use a response file for the Configuration Wizard, see the MicroStrategy 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 Wizard uses. For more information about how to set up and use a `setup.iss` file for a silent MicroStrategy installation, see the MicroStrategy 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 of the security implementations listed below are described in detail in the preceding sections of this chapter.

<table>
<thead>
<tr>
<th>Security implementation</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Set up access controls for the database (see Controlling access to data, page 88). Depending on your security requirements you may need to:</td>
<td></td>
</tr>
<tr>
<td>• Set up security views to restrict access to specific tables, rows, or columns in the database</td>
<td></td>
</tr>
<tr>
<td>• 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</td>
<td></td>
</tr>
<tr>
<td>• Implement connection mapping to control individual access to the database</td>
<td></td>
</tr>
<tr>
<td>• Configure passthrough execution to control individual access to the database from each project, and to track which users are accessing the RDBMS system</td>
<td></td>
</tr>
<tr>
<td>• Assign security filters to users or groups to control access to specific data (these operate similarly to security views, but at the application level)</td>
<td></td>
</tr>
<tr>
<td>Understand the MicroStrategy user model (see The MicroStrategy user model, page 62). Use this model to:</td>
<td></td>
</tr>
<tr>
<td>• Select and implement a system authentication mode to identify users</td>
<td></td>
</tr>
<tr>
<td>• Set up security roles for users and groups to assign basic privileges and permissions</td>
<td></td>
</tr>
<tr>
<td>• Understand ACLs (access control lists) which allow users access permissions to individual objects</td>
<td></td>
</tr>
<tr>
<td>• 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.</td>
<td></td>
</tr>
<tr>
<td>• If you leave anonymous access disabled, ensure the hyperlink for guest access does not appear on the login page. Use adminoptions.asp to turn this hyperlink off.</td>
<td></td>
</tr>
<tr>
<td>Assign privileges and permissions to control user access to application functionary. You may need to:</td>
<td></td>
</tr>
<tr>
<td>• 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</td>
<td></td>
</tr>
<tr>
<td>• Make sure guest users (anonymous authentication) have access to the Log folder located in C:\Program Files\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.</td>
<td></td>
</tr>
</tbody>
</table>
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 particularly sensitive or confidential data, enable the setting to encrypt all communication between Web server and Intelligence Server. **Note**: There may be a noticeable performance degradation since the system must encrypt and decrypt all network traffic.

Locate the physical machine hosting the Web application in a physically secure location.

Restrict access to files stored on the machine hosting the 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 /MicroStrategy/admin/admin.asp.) Be sure to restrict access to:

- The Admin directory
- adminoptions.asp
- delete.asp
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 62
- Controlling access to application functionality, page 69
- Controlling access to data, page 88
- Merging users or groups, page 113

Authentication, the process by which the system identifies the user, is an integral part of any security model. Authenticating users is addressed in Chapter 3, Identifying Users: Authentication.
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 169.

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 Desktop Preferences dialog box, in the Desktop: 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 particular 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:

- The Everyone group, page 63
- Authentication-related groups, page 64
- Groups corresponding to product offerings, page 64
- Administrator groups, page 65

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 Desktop privilege is automatically granted to the Everyone group. This ensures that all users who were able to access Desktop 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 3, 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 122.

- **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 126.

- **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.

- **Desktop Analyst**: Desktop 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.

- **Desktop Designer**: Desktop Designers 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 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.

For detailed information about privileges, including how to assign privileges to a user or group, see *Controlling access to functionality: Privileges, page 79*. 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 226*. 
To view a user’s privileges

1. In Desktop, 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 Desktop.

3. Right-click the user and select Project Access. The User Editor opens. 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 69.

To view the permissions for an object

1. From within MicroStrategy Desktop, 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.
Creating, importing, and deleting users and groups

It is possible to create users individually using the User Manager interface in MicroStrategy Desktop, or using the Command Manager utility that is part of Administrator. 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 Desktop

1. In Desktop, 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 tab. For details about each field, see the Desktop 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 Desktop 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 Desktop, 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:

- 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 Desktop, 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 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 88), 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 69
• Controlling access to functionality: Privileges, page 79
• Defining sets of privileges: Security roles, page 82

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 a special privilege called Bypass All Object Security Access Checks. Users with this privilege are not restricted by access control permissions and are considered to have full control over all objects. For information about privileges, see Controlling access to functionality: Privileges, page 79.

To modify permissions for an object in Desktop

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

   To modify an object's ACL, you must access the Properties dialog box directly from Desktop. 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.

<table>
<thead>
<tr>
<th>Grouping</th>
<th>Description</th>
<th>Permissions granted</th>
</tr>
</thead>
</table>
| 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 |
| 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. **Note:** This permission overrides any permissions the user may inherit from any other sources. | none; all are denied |
| Default    | Neither grants nor denies permissions; all permissions are inherited from the underlying group. | none |
| Custom     | Allows the user or group to have a custom combination of permissions that you can define. | custom choice |
The permissions actually assigned to the user or group when you select a permission grouping are explained in the table below.

<table>
<thead>
<tr>
<th>Grouping</th>
<th>Description</th>
<th>Permissions granted</th>
</tr>
</thead>
</table>
| 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 |

<table>
<thead>
<tr>
<th>Permission</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse</td>
<td>View the object in Desktop and Web</td>
</tr>
<tr>
<td>Read</td>
<td>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.</td>
</tr>
<tr>
<td>Write</td>
<td>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.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the object</td>
</tr>
<tr>
<td>Control</td>
<td>Modify the object’s access control list</td>
</tr>
</tbody>
</table>
| 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 77. When applied to a language object, allows users to edit and save translations, and to select the language for display in their Desktop or Web language preferences. This permission is checked at design time.  
**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 77. This permission is checked at run time. |
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 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.

<table>
<thead>
<tr>
<th>Grouping</th>
<th>Permissions Granted</th>
<th>Allows you to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect</td>
<td>• Browse</td>
<td>Connect to the server</td>
</tr>
<tr>
<td>Monitoring</td>
<td>• Browse</td>
<td>• View server definition properties</td>
</tr>
<tr>
<td></td>
<td>• Read</td>
<td>• View statistics settings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the system monitors</td>
</tr>
</tbody>
</table>
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 Checks allows the user to ignore the access checks for schema objects.
  - Bypass All Object Security Checks allows the user to ignore the access checks for all objects.
Permission levels

If a user has permissions assigned to an object and is in a group that has a different permission grouping assigned to the object, the highest level of permission is granted to the user.

Permissions ranked from highest level down to lowest level are listed below. The permissions at the top of the list override those permissions lower down the list, when both types of permissions are assigned to a user:

1. Denied All: Highest level permission
2. The permissions with the fewest restrictions
3. The permissions with the most restrictions (except Denied All): Lowest level permissions

For example, if a user has Full Control permissions for a report, and is a member of the Managers group, which has View permissions for the report, the user has Full Control permissions for the report. If the user later becomes a member of a group which has the Denied All permission for the report, the user has no permissions at all for the report.

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
  - 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 only at object design time.
- 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.
- Use permission only: The user can create reports using the object, but cannot execute those reports.
- 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.
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 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 Desktop, 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 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 461.

**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, 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 69.

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 226).

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 82).

To assign privileges to users or groups

1  From MicroStrategy Desktop User Manager, edit the user with the User Editor or edit the group with the Group Editor.

2  Click the Project Access tab.

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 82.

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.

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 Desktop, 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 Desktop Help (press F1 from within the Find and Replace Privileges dialog box).
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 79)
- Privileges assigned to any groups of which the user is a member (see About MicroStrategy user groups, page 63)
  
  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 82)
- 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 63. 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 64.
  
  International Users is a member of the following product-based groups: Desktop Analyst, Mobile User, Web Reporter, and Web Viewer. 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.
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 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 Web privileges.

- **Desktop Analyst**
  - **Desktop Designer**

  In the case of Desktop user groups, the Desktop Designer inherits the privileges of the Desktop Analyst and therefore has more privileges than the Desktop 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 Desktop Analyst, Mobile User, Web Reporter, and Web Viewer 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 81*.

**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 85*.

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

1. In Desktop, 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 Desktop 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.

**To create a security role**

1 In Desktop, 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.
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 83*.

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 Desktop Help. (From within Desktop, press F1.)

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 12, 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 Desktop, right-click on the project you want to deny access to. Select Project Configuration. The Project Configuration Editor opens.
2 Expand **Project Access**. The Project Access - General dialog box opens.

3 From 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 Desktop, 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 Desktop. The User Editor opens.

9 On the Project Access tab, 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.

10 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 pre-defined security roles for administrators. These roles makes it easy to delegate administrative tasks.

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 Checks. None of the roles have any privileges in common. For a list of the privileges included with each pre-defined 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 all privileges for a project.

- **Project Bulk Administrators**, who can perform administrative functions on multiple objects with Object Manager (see Copying objects between projects: Object Manager, page 256), Command Manager (see Chapter 12, 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 83.

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).

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

- Controlling access to the database: Connection mappings, page 88
- Linking database users and MicroStrategy users: Passthrough execution, page 93
- Restricting access to data: Security filters, page 95
- Controlling access to data at the database (RDBMS) level, page 109

Controlling access to the database: Connection mappings

Connection mappings allow you to assign a user or group in the MicroStrategy system to a specific 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 109.

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.

Creating a connection mapping

You define connection mappings with the Project Configuration Editor in MicroStrategy Desktop. 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 9.

To create a connection mapping

1 In Desktop, 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 the Connection Mapping category.

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.”

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 9. For information on creating a new database login, see Connecting to the data warehouse, page 9.

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:

- “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 123*), but works for other authentication modes as well.

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?

There are several reasons for using passthrough execution:

- **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 on a per-account basis. 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 MicroStrategy Desktop. 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 9*.

**To enable linked warehouse logins**

1. In Desktop, 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, then Authentication, and then select Warehouse.

4. Select the **Use warehouse passthrough credentials** check box.

5. To use warehouse credentials for all database instances, select the **For all database instances** option.

6. To use warehouse credentials only 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 your changes are saved.

8 To specify the login

## 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 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 109.

For more information about security filters, see the following:

- *Security filter example*, page 95
- *How security filters work*, page 96
- *Creating and applying a security filter*, page 97
- *Security filters and metric levels*, page 98
- *Using a single security filter for multiple users: System prompts*, page 107
- *Merging security filters*, page 104

### 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 in the example below.

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Item</th>
<th>Metrics</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics</td>
<td>TV's</td>
<td>GPX 5&quot; AM/FM Portable TV</td>
<td></td>
<td>$76,703</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 32&quot; Stereo TV</td>
<td></td>
<td>$436,830</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA Indoor TV Antenna</td>
<td></td>
<td>$32,781</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA Power TV Antenna</td>
<td></td>
<td>$73,040</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 27&quot; Stereo TV</td>
<td></td>
<td>$236,160</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 13&quot; TV/VCR</td>
<td></td>
<td>$168,688</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 13&quot; TV</td>
<td></td>
<td>$92,307</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 4&quot; LCD Color TV</td>
<td></td>
<td>$156,090</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 2&quot; Diagonal LCD Color TV</td>
<td></td>
<td>$145,235</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 25&quot; TV/VCR Combo</td>
<td></td>
<td>$271,852</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 25&quot; Stereo Color TV</td>
<td></td>
<td>$208,474</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 13&quot; 2-Head TV/VCR</td>
<td></td>
<td>$168,947</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 32&quot; Color TV</td>
<td></td>
<td>$460,395</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sony 32&quot; Trinitron Television</td>
<td></td>
<td>$566,960</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sony 35&quot; Trinitron Television</td>
<td></td>
<td>$703,446</td>
</tr>
</tbody>
</table>

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 in the example below. The user cannot see any data from attribute elements that are outside the security filter.

<table>
<thead>
<tr>
<th>Category</th>
<th>Metrics</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics</td>
<td></td>
<td>$3,837,906</td>
</tr>
</tbody>
</table>

**How security filters work**

Security filters are the same as regular filters except that they can only contain 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 MicroStrategy 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. This is also true for element browsing: when the user browses through a hierarchy to answer a prompt, he or she will only see the attribute elements as defined by the security filter.

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 9, Improving Report and Document Response Time: Caching.

Each user or group can be directly assigned only one security filter for a given 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 security filters may need to be merged. For information about how security filters are merged, see Merging security filters, page 104.

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 Desktop Designer 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 Desktop, 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 Desktop 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.

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

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

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 102*.

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.

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

**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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Item</th>
<th>Metrics</th>
<th>Revenue</th>
<th>Category Revenue</th>
<th>Subcategory Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics</td>
<td>TVs</td>
<td>GPX 5&quot; AM/FM Portable TV</td>
<td>$76,703</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 32&quot; Stereo TV</td>
<td>$436,830</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA Indoor TV Antenna</td>
<td>$32,781</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA Power TV Antennas</td>
<td>$73,040</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 27&quot; Stereo TV</td>
<td>$236,160</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 13&quot; TV/VCR</td>
<td>$188,888</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 13&quot; TV</td>
<td>$92,307</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 4&quot; LCD Color TV</td>
<td>$156,090</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 2&quot; Diagonal LCD Color TV</td>
<td>$145,235</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 25&quot; TV/VCR Combo</td>
<td>$271,852</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 25&quot; Stereo Color TV</td>
<td>$208,474</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 13&quot; 2-Head TV/VCR</td>
<td>$168,947</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 32&quot; Color TV</td>
<td>$460,395</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sony 32&quot; Trinitron Television</td>
<td>$586,960</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sony 35&quot; Trinitron Television</td>
<td>$703,446</td>
<td>$3,837,905.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
</tbody>
</table>

**Bottom range attribute: Subcategory**

If a bottom range attribute is specified, then 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, since 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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Item</th>
<th>Metrics</th>
<th>Revenue</th>
<th>Category Revenue</th>
<th>Subcategory Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics</td>
<td>TV's</td>
<td>GPX 5” AM/FM Portable TV</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 32” Stereo TV</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA Indoor TV Antenna</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA Power TV Antenna</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 27” Stereo TV</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 13” TV/VCR</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 13” TV</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 4” LCD Color TV</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCA 2” Diagonal LCD Color TV</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 25” TV/VCR Combo</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 25” Stereo Color TV</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,906.86</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 13” 2-Head TV/VCR</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharp 32” Color TV</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sony 32” Trinitron Television</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sony 35” Trinitron Television</td>
<td>$3,837,906</td>
<td>$24,391,302.85</td>
<td>$3,837,905.85</td>
<td></td>
</tr>
</tbody>
</table>

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 on a per-user basis.

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 Desktop, 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 104** (This is the default method for merging security filters)
- **Merging all security filters with AND, page 106**

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

<table>
<thead>
<tr>
<th>Group</th>
<th>Security Filter</th>
<th>Hierarchy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics</td>
<td>Category = Electronics</td>
<td>Product</td>
</tr>
<tr>
<td>Drama</td>
<td>Subcategory = Drama</td>
<td>Product</td>
</tr>
<tr>
<td>Movies</td>
<td>Category = Movies</td>
<td>Product</td>
</tr>
<tr>
<td>Northeast</td>
<td>Region = Northeast</td>
<td>Geography</td>
</tr>
</tbody>
</table>

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 Security Filters 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 472*.

**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:

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Metrics</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics</td>
<td>Audio Equipment</td>
<td>$3,782,832</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cameras</td>
<td>$5,061,148</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computers</td>
<td>$1,926,998</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Electronics - Miscellaneous</td>
<td>$4,871,957</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TV's</td>
<td>$3,837,906</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Video Equipment</td>
<td>$5,108,464</td>
<td></td>
</tr>
<tr>
<td>Movies</td>
<td>Drama</td>
<td>$698,840</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Region</th>
<th>Category</th>
<th>Subcategory</th>
<th>Metrics</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Northeast</td>
<td>Movies</td>
<td>Action</td>
<td>$148,648</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Comedy</td>
<td>$164,441</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drama</td>
<td>$168,547</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Horror</td>
<td>$155,926</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kids / Family</td>
<td>$162,235</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special Interests</td>
<td>$201,764</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Metrics</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movies</td>
<td>Drama</td>
<td></td>
<td>$638,840</td>
</tr>
</tbody>
</table>

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 Desktop, log into a project.
2 From the Administration menu, point to Projects, and then select Project Configuration. The Project Configuration Editor opens.
3 Expand the Security Filters category and select the General subcategory.
4 Under Rules for merging multiple security filters, 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 104)

- Intersect (AND) all security filters (see Merging all security filters with AND, page 106)

5 Click OK to close the Project Configuration Editor. You must restart Intelligence Server for your changes to take effect.

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 located in the Public Objects/Prompts/System Prompts folder in Desktop.

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 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 166.

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

- Simplifying the security filter definition process, page 108
- Implementing a report-level security filter, page 109
- Using database tables that contain security information, page 109
To create a security filter using a system prompt

1. In Desktop, 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.


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.

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 MicroStrategy 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 11351.

**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 110**
- **Splitting fact tables by rows, page 110**
- **Splitting fact tables by columns, page 112**

### 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 particular 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:

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

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:

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Customer Address</th>
<th>Member Bank</th>
<th>Transaction Amount ($)</th>
<th>Current Balance ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456</td>
<td>12 Elm St.</td>
<td>1st National</td>
<td>400.80</td>
<td>40,450.00</td>
</tr>
<tr>
<td>886580</td>
<td>907 Grove Rd.</td>
<td>1st National</td>
<td>76.35</td>
<td>10,333.45</td>
</tr>
</tbody>
</table>

The table for Eastern Credit would look like this:

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Customer Address</th>
<th>Member Bank</th>
<th>Transaction Amount ($)</th>
<th>Current Balance ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>945940</td>
<td>888 Oak St.</td>
<td>Eastern Credit</td>
<td>150.00</td>
<td>60,010.70</td>
</tr>
<tr>
<td>562055</td>
<td>1 Ocean Blvd.</td>
<td>Eastern Credit</td>
<td>888.50</td>
<td>1,000.00</td>
</tr>
</tbody>
</table>

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.

**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:

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Customer Address</th>
<th>Member Bank</th>
<th>Transaction Amount ($)</th>
<th>Current Balance ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Customer Address</th>
<th>Member Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Transaction Amount ($)</th>
<th>Current Balance ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|
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 113
• Running the User Merge Wizard, page 116

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 255.

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.

**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 particular project, the destination user does not have a security role on that project.
- If the destination user has no security role for a particular 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.

**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 **Programs**, then **MicroStrategy**, then **Object Manager**, 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 113*.

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.
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.

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 120*.

- **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 122*.

- **Database warehouse**: The data warehouse database is the authentication authority. For more information, see *Implementing database warehouse authentication, page 123*.

- **LDAP (lightweight directory access protocol)**: An LDAP server is the authentication authority. For more information, see *Implementing LDAP authentication, page 126*.

- **Single sign-on**: Single sign-on encompasses several different third-party authentication methods, including:
  - **Windows authentication**: Windows is the authentication authority. For more information, see *Implementing Windows NT authentication, page 169*.
  - **Integrated authentication**: A domain controller using Kerberos authentication is the authentication authority. For more information, see *Enabling integrated authentication, page 176*.
  - **Third-party authentication**: A third-party single sign-on tool, such as Tivoli, SiteMinder, or Oblix, is the authentication authority. For more information, see *Enabling Single Sign-on to Web, Mobile, and Office with third-party authentication, page 199*.

For examples of situations where you might want to implement specific authentication modes, and the steps to do so, see *Authentication examples, page 217*.

**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 120*).
To configure the authentication mode for a project source

1. In Desktop, 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.)
   - 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 9. In addition, standard authentication is the only authentication mode that allows a user or system administrator to change or expire MicroStrategy passwords.

When using standard authentication, the MicroStrategy 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 Desktop Help. (From within Desktop, press F1.)

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 Desktop Help. (From within Desktop, press F1.)

High-level steps to configuration standard authentication

1. In Desktop, 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.)

3 In MicroStrategy 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 In Desktop, 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 79* and *Controlling access to objects: Permissions, page 69*.

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 Desktop, log into the project source with a user that has administrative privileges.

2. From the Desktop Folder List, select Administration.

3. From the File menu, select Properties. The Properties - project source name dialog box opens.


5. Select the Public/Guest group.

6. In the Access Permission list, select Connect.

7. Click OK. The Select Desktop 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 118 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 MicroStrategy Intelligence Server and an LDAP server when an LDAP user logs into Desktop 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 122*.

   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 Desktop Help. (From within Desktop, press F1.)

**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 120*).

3. Configure the project source to use database warehouse authentication (see *Configuring the authentication mode for a project source, page 118*).

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 83*).

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 67*.]

6. For each project, in the Project Configuration Editor, in the 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 93*).
To enable database authentication in MicroStrategy Web, access the Preferences: Project defaults: Login page and select the Database Authentication check box, 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 125.

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 excels in 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.

When using LDAP authentication, LDAP users and groups are either linked directly to MicroStrategy users and groups, or they are imported into the MicroStrategy metadata. For information on the difference between importing and linking LDAP users and groups and steps to implement each setup in MicroStrategy, see *Authenticating LDAP users and groups in MicroStrategy, page 139.*

You can also set up MicroStrategy Office to use LDAP authentication. For information, see the *MicroStrategy Office User's Guide.*

**LDAP information flow**

The following scenario presents an overview of the general flow of information between MicroStrategy Intelligence Server and an LDAP server when an LDAP user logs into Desktop or MicroStrategy Web:

1. When an LDAP user logs in to MicroStrategy Web or Desktop, Intelligence Server connects to the LDAP server using the authentication user (the main administrative user). (For instructions on how to configure this connection, see *Implementing LDAP authentication in MicroStrategy, page 128.*)

2. The authentication user is bound to LDAP using this user’s 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 Desktop 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. For more information on linked users, see *Linking users and groups without importing, page 152.*

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. For more
Implementing LDAP authentication in MicroStrategy

The following procedures must be completed to set up the minimal requirements to implement LDAP authentication in MicroStrategy:

- **Setting up LDAP SDK connectivity, page 129**—Install and configure your LDAP SDK files with Intelligence Server.

- **Defining LDAP connection parameters, page 132**—Define the parameters that are required to create a connection between your LDAP server and MicroStrategy.

- **Implementing LDAP authentication for a project source, page 134**—Set up user authentication in MicroStrategy based on your LDAP directory for a specific project source.

- **Configuring LDAP authentication, page 134**—step through the process of creating a connection between your LDAP server and MicroStrategy using the LDAP Connectivity Wizard.

Once you have configured your LDAP server’s connection with your MicroStrategy system, use the following procedures to configure your MicroStrategy system to use LDAP authentication.

- **Defining LDAP search filters to verify and import users and groups at login, page 135**—provide Intelligence Server with the information necessary to search your LDAP directory for users and groups.

- **Authenticating LDAP users and groups in MicroStrategy, page 139**—choose whether to import LDAP users and groups into your MicroStrategy system, to link LDAP users to existing MicroStrategy...
users, or to allow LDAP users to log in to your MicroStrategy system as anonymous guest users.

- *Synchronizing imported LDAP users and groups, page 154*— keep your MicroStrategy and LDAP user information up to date.

- *Managing LDAP authentication in MicroStrategy, page 159*— implement different LDAP security models, enable connection pooling to potentially improve performance, and use LDAP attributes to create security filters.

### 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 MicroStrategy 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.

1: The behavior between Intelligence Server and the LDAP SDK varies slightly depending on the LDAP SDK used. The MicroStrategy readme lists recommendations.

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 Desktop 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 to the system environment variable so that Intelligence Server can locate them.

- UNIX/Linux environment: Modify the `LDAP.sh` file located in the `/Installations/env` folder to point to the location of the LDAP SDK libraries. The detailed procedure is described in the procedure "To add the library path to the environment variable in UNIX" below.

4 Restart Intelligence Server.

---

**To add the library 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](#).

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 connection parameters

To connect your LDAP server to Intelligence Server, there is a minimum set of parameters that must be configured. These parameters must be configured after setting up your LDAP SDK, which is covered in the section Setting up LDAP SDK connectivity, page 129. To define the required connection parameters, follow the steps below.

To define the required LDAP configuration parameters

1. In Desktop, log in to 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 LDAP category and then select Server.

4. Type the information for the Host field and choose the appropriate Port.
   - **Host**
     The LDAP host is either the host machine name or IP address of the host machine of the LDAP server.
   - **Port**
     The LDAP port is the port number of the LDAP server. Port 389 is the default when connecting with clear text, and port 636 is the default for SSL. However, the LDAP port can be set to a different number than the default. Confirm the accurate port number with your LDAP administrator.

5. Select whether to use a Clear Text or SSL (encrypted) security connection.
   - **Connecting with clear text or encrypted SSL**
     Intelligence Server can connect to an LDAP server either using clear text or by transmitting encrypted information using SSL. MicroStrategy recommends using SSL to protect information transmitted between Intelligence Server and the LDAP server.
If you will be implementing SSL encryption, use the following steps to set it up:

**High-level steps to use SSL connectivity**

Make sure you have completed the procedure *High-level steps to install the LDAP SDK DLLs, page 130* to set up the LDAP SDK DLLs so that the LDAP file location is registered before setting up SSL connectivity.

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.

c. In MicroStrategy Desktop, configure the LDAP SSL certificate settings in the Intelligence Server Configuration Editor to point to the certificate. For steps to access the Intelligence Server Configuration Editor, see *To define the required LDAP configuration parameters, page 132*. Depending on your LDAP server vendor, you 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.

d. Select the appropriate settings for the LDAP server vendor name, connectivity driver, connectivity files, and Intelligence Server platform. For details on setting these connection parameters, see *Setting up LDAP SDK connectivity, page 129*. 
Implementing LDAP authentication © 2011 MicroStrategy, Inc.

Type the appropriate information in the **Distinguished name (DN)** and **Password** fields for the authentication user.

- **Authentication user**

  The main administrative user, referred to as the authentication user, is used by Intelligence Server to access, search in, and retrieve information from the LDAP server when authenticating, importing, and synchronizing new user accounts.

  No MicroStrategy user from the metadata is associated with the authentication user, and only a temporary session is created for this type of authentication. However, the authentication user does appear in the MicroStrategy Connection Monitor.

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

Implementing LDAP authentication for a project source

To implement LDAP authentication in MicroStrategy, you must implement LDAP authentication for each project source that you want to use LDAP authentication for.

**To implement LDAP authentication for a project source**

1. In MicroStrategy Desktop, right-click the project source, and select **Modify Project Source**. The Project Source Manager opens.

2. On the **Advanced** tab, select **Use LDAP Authentication**.

3. Click **OK** to accept your changes and close the Project Source Manager.

Configuring LDAP authentication

The LDAP Connectivity Wizard steps you through the different required and optional connection parameters to implement LDAP authentication in MicroStrategy.
To configure LDAP authentication with the LDAP Connectivity Wizard

1. In MicroStrategy Desktop, 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 LDAP Connectivity Wizard. The LDAP Connectivity Wizard opens.

3. Step through the pages of the wizard. For information about each page of the wizard, click Help.

4. On the Summary page, click Finish to create and save your LDAP connection.

Upon completing 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.

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.

To set the search parameters, you enter these parameters in the Intelligence Server Configuration Editor, in the User Search Filter and Group Search Filter dialog boxes. To access the Intelligence Server Configuration Editor, right click a project source. Detailed steps for how to enter this information can be found in the Desktop Help.

A Distinguished Name is a unique way to define a user or group within the LDAP directory structure. Intelligence Server uses the Distinguished Name of users and groups to authenticate them and to link them to existing MicroStrategy users and groups. MicroStrategy also provides a configuration option to import LDAP users and groups in their Distinguished Name format. For more information on importing LDAP users and groups into MicroStrategy, see Authenticating LDAP users and groups in MicroStrategy, page 139.

If you do not provide the search root, the user search filter, and the group search filter searches of the LDAP directory might perform poorly. Highest level to start an LDAP search: Search root, page 136, provides examples of
these parameters as well as additional details of each parameter and some LDAP server-specific notes.

**Highest level to start an LDAP search: Search root**

To search effectively, Intelligence Server must know where to start its search. When setting up LDAP authentication, 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.

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.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Search Root</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include all users and groups from Operations</td>
<td>Operations</td>
</tr>
<tr>
<td>Include all users and groups from Operations, Consultants, and Sales</td>
<td>Sales</td>
</tr>
<tr>
<td>Include all users and groups from Operations, Consultants, and Technology</td>
<td>Departments</td>
</tr>
<tr>
<td>Include all users and groups from Technology and Operations but not Consultants.</td>
<td>Departments (with an exclusion clause in the User/Group search filter to exclude users who belong to Consultants.)</td>
</tr>
</tbody>
</table>
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=labs, dc=micrstrategy, dc=com”:

![Diagram showing LDAP structure]

The search root parameter searches for users in the “leaves” of the “tree” who are all registered within a single domain. 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. These filters can be configured in the Intelligence Server Configuration Editor, under the LDAP subject.

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 *About group search filters, page 138*).
User search filters are generally in the form

\[(\&(\text{objectclass=LDAP\_USER\_OBJECT\_CLASS})\ (\text{LDAP\_LOGIN\_ATTR}=#\text{LDAP\_LOGIN#}))\text{ where:}\]

- \text{LDAP\_USER\_OBJECT\_CLASS} indicates the object class of the LDAP users. For example, you can enter
  \[(\&\text{(objectclass=person}) (\text{cn}=#\text{LDAP\_LOGIN#}))\].

- \text{LDAP\_LOGIN\_ATTR} indicates which LDAP attribute to use to store LDAP logins. For example, you can enter
  \[(\&\text{(objectclass=person}) (\text{cn}=#\text{LDAP\_LOGIN#}))\].

- \#\text{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, \(\{(\&\text{(objectclass=person}) (\text{uniqueID}=#\text{LDAP\_LOGIN#}))\}\), where \text{uniqueID} is the LDAP attribute name your company uses for authentication.

**About 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 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):

- \(\{(\&\text{(objectclass=LDAP\_GROUP\_OBJECT\_CLASS}) (\text{LDAP\_MEMBER\_LOGIN\_ATTR}=#\text{LDAP\_LOGIN#}))\}\)

- \(\{(\&\text{(objectclass=LDAP\_GROUP\_OBJECT\_CLASS}) (\text{LDAP\_MEMBER\_DN\_ATTR}=#\text{LDAP\_DN#}))\}\)

- \(\{(\&\text{(objectclass=LDAP\_GROUP\_OBJECT\_CLASS}) (\text{gidNumber}=#\text{LDAP\_GIDNUMBER#}))\}\)
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.

**Authenticating LDAP users and groups in 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.

<table>
<thead>
<tr>
<th>Connection Type</th>
<th>Benefits</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Importing LDAP users and groups into MicroStrategy</td>
<td>• Users and groups are created in the metadata.</td>
<td>• In environments that have many LDAP users, importing can quickly fill the metadata with these users and their related information.</td>
</tr>
<tr>
<td></td>
<td>• Users and groups can be assigned additional privileges and permissions in MicroStrategy.</td>
<td>• Users and groups may not have the correct permissions and privileges when they are initially imported into MicroStrategy.</td>
</tr>
<tr>
<td></td>
<td>• Users have their own inboxes and personal folders in MicroStrategy.</td>
<td></td>
</tr>
<tr>
<td>Linking users and groups without importing</td>
<td>• For environments that have many LDAP users, linking avoids filling the metadata with users and their related information.</td>
<td>• Users to be linked to must already exist in the MicroStrategy metadata.</td>
</tr>
<tr>
<td></td>
<td>• You can use Command Manager to automate the linking process using scripts. See the Command Manager Help for details.</td>
<td></td>
</tr>
<tr>
<td>Allowing anonymous/guest users with LDAP authentication</td>
<td>• Users can log in immediately without having to create a new MicroStrategy user.</td>
<td>• Privileges are limited to those for the Public/Guest group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Users’ personal folders and Inboxes are deleted from the system after they log out.</td>
</tr>
</tbody>
</table>

If your environment includes multiple Intelligence Servers connected to one MicroStrategy Web server, see MicroStrategy Tech Note TN11151 for details on setting up LDAP authentication.

When you import LDAP users and groups in MicroStrategy, all authenticated LDAP users and groups are imported into the MicroStrategy metadata, which means that a physical MicroStrategy user is created within the MicroStrategy metadata. For further details about and steps for implementing this option, see *Importing LDAP users and groups into MicroStrategy, page 141*.

Another option is to link LDAP users and groups to existing users and groups in MicroStrategy. This helps keep the metadata from filling with data related to your LDAP users. However, it does require that the users to be linked exist in the metadata. For further details about and steps for implementing this option, see *Linking users and groups without importing, page 152*.

If you choose to not import or link LDAP users and groups, no LDAP users or groups are imported into the metadata when LDAP authentication occurs. Instead, you can configure the system so that LDAP users can log into
MicroStrategy as anonymous users, who receive the privileges of the MicroStrategy Public/Guest group. In this case the users’ personal folders and Inboxes do not persist once they log out of the MicroStrategy system. For details on this authentication option, see *Allowing anonymous/guest users with LDAP authentication, page 153.*

**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. For information on setting up user and group import options, see the following sections:

- *Importing users and groups at login, page 142*
- *Importing a list of users and groups in batch, page 143*

Imported users are automatically a member 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 MicroStrategy Desktop. 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:

<table>
<thead>
<tr>
<th>Desktop Login</th>
<th>Name</th>
<th>Owner</th>
<th>Account Status</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finance Group</td>
<td>Finance Group</td>
<td>Administrator</td>
<td></td>
<td>8/23/200...</td>
</tr>
<tr>
<td>jdoe</td>
<td>John Doe</td>
<td>Administrator</td>
<td>Enabled</td>
<td>8/23/200...</td>
</tr>
<tr>
<td>msmith</td>
<td>Mary Smith</td>
<td>Administrator</td>
<td>Enabled</td>
<td>8/23/200...</td>
</tr>
<tr>
<td>Sales Group</td>
<td>Sales Group</td>
<td>Administrator</td>
<td></td>
<td>8/23/200...</td>
</tr>
</tbody>
</table>

If you want a user’s group 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. For more information on specifying an authentication user’s DN and password, see Defining LDAP connection parameters, page 132.

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.

Importing users and groups at login

You can choose to import users and their associated groups when a user logs in to MicroStrategy for the first time. 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.

Once a user or group is created in MicroStrategy, you can assign privileges and security settings that control what a user can access in MicroStrategy. For more information on assigning security settings, see User privileges and security settings after import, page 146. Users imported into MicroStrategy are also given their own inboxes and personal folders.

To import users and/or groups at login

1 In MicroStrategy Desktop, 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 Import/Synchronize.

4 In the Import/Synchronize at Login area:
   - Select the Import Users check box to import users at login.
   - Select the Synchronize MicroStrategy User Login/User Name with LDAP check box to allow MicroStrategy’s user information to synchronize with the LDAP user information. For more information, see Selecting the LDAP attributes to import for user and group information, page 147.
   - Select the Import Groups check box to import groups at login.
   - Select the Synchronize MicroStrategy Group Name with LDAP check box to allow MicroStrategy’s group information to synchronize with the LDAP group information. For more information, see Selecting the LDAP attributes to import for user and group information, page 147.

The user names, user logins, and group names that are imported into MicroStrategy depend on the LDAP attributes you select to import. To designate the LDAP attributes that MicroStrategy will import, see the following sections:

- Selecting the LDAP attributes to import for user and group information, page 147
- Importing integrated authentication credentials associated with LDAP users, page 149
- Importing email addresses associated with LDAP users, page 150
- Selecting how many nested groups to import with the user, page 150

Importing a list of users and groups in batch

You can import a list of users and their associated groups in batch. 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.
Once a user or group is created in MicroStrategy, you can assign privileges and security settings that control what a user can access and do in MicroStrategy. Therefore, by importing users and groups in batch, you can assign privileges and security settings for those users and groups before the users log in to MicroStrategy. This allows you to setup system security ahead of time.

Users imported into MicroStrategy are also given their own inboxes and personal folders.

To allow users and groups to be imported in batch

1. In MicroStrategy Desktop, 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 Import/Synchronize. The Import/Synchronize options are displayed.

4. In the Import/Synchronize in Batch area:
   - To import users in batch, select the Import Users check box. 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. For more information on defining a user search filter to import a list of users, see Defining a search filter to return a list of users, page 145.
   - To synchronize MicroStrategy’s user information with the LDAP user information, select the Synchronize MicroStrategy User Login/User Name with LDAP check box. For more information, see Synchronizing imported LDAP users and groups, page 154.
   - To import groups in batch, select the Import Groups check box. 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. For more information on defining a user search filter to import a list of users, see Defining a search filter to return a list of groups, page 146.
   - To synchronize MicroStrategy’s group information with the LDAP group information, select the Synchronize MicroStrategy Group Name with LDAP check box. For more information, see Synchronizing imported LDAP users and groups, page 154.
The user names, user logins, and group names that are imported into MicroStrategy depend on the LDAP attributes you select to import. For more information see the following sections:

- *Selecting the LDAP attributes to import for user and group information, page 147*
- *Importing integrated authentication credentials associated with LDAP users, page 149*
- *Importing email addresses associated with LDAP users, page 150*
- *Selecting how many nested groups to import with the user, page 150*

**Defining a search filter to return a list of users**

If you choose to import or synchronize a list of users in batch, you must provide a user search filter to return a list of users. You should contact your LDAP administrator for the proper user search filter syntax. A user search filter is generally of the following form:

```
(& (objectclass=LDAP_USER_OBJECT_CLASS) (LDAP_LOGIN_ATTR=SEARCH_STRING))
```

The user search filter form given above has the following placeholders:

- **LDAP_USER_OBJECT_CLASS** indicates the object class of the LDAP users. For example, you can enter `(& (objectclass=person) (cn=h*))`.

- **LDAP_LOGIN_ATTR** indicates which LDAP attribute to use to store LDAP logins. For example, you can enter `(& (objectclass=person) (cn=h*))`.

- **SEARCH_STRING** indicates the search criteria for your user search filter. You must match the correct LDAP attribute for your search filter. For example, you can search for all users with an LDAP user login that begins with the letter h by entering `(& (objectclass=person) (cn=h*))`.

Depending on your LDAP server vendor and your tree structure created within LDAP, you may need to try different attributes within the search filter syntax above. For example:

```
(& (objectclass=person) (uniqueID=SEARCH_STRING))
```

where **uniqueID** is the LDAP attribute your company uses for authentication.
For information on how and where to enter your user search filters for importing and synchronizing users in batch, see *To allow users and groups to be imported in batch, page 144.*

**Defining a search filter to return a list of groups**

Enter a group search filter to return a list of groups to import in batch. You should contact your LDAP administrator for the proper group search filter syntax. A group search filter is generally of the following form:

\[
(& (objectclass=LDAP\_GROUP\_OBJECT\_CLASS) (LDAP\_GROUP\_ATTR=SEARCH\_STRING))
\]

The group search filter form given above has the following placeholders:

- **LDAP\_GROUP\_OBJECT\_CLASS** indicates the object class of the LDAP groups. For example, you can enter 
  \[(& (objectclass=groupOfNames) (cn=h*))\].

- **LDAP\_GROUP\_ATTR** indicates which LDAP attribute of an LDAP group is searched for to retrieve a list of groups. For example, you can enter 
  \[(& (objectclass=groupOfNames) (cn=h*))\].

- **SEARCH\_STRING** indicates the search criteria for your group search filter. You must match the correct LDAP attribute for your search filter. For example, you can search for all groups with an LDAP group name that begins with the letter h by entering 
  \[(& (objectclass=groupOfNames) (cn=h*))\].

For information on how and where to enter your group search filters for importing and synchronizing groups in batch, see *To allow users and groups to be imported in batch, page 144.*

**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 Desktop 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 displays all privileges for each project in the project source. For details on editing user and group privileges, see the Desktop 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. For more information, see Synchronizing imported LDAP users and groups, page 154.

Selecting the LDAP attributes to import for user and group information

When importing LDAP users as MicroStrategy users, you must choose which LDAP attributes are imported as the MicroStrategy user login, user name, and group name.

• The user login is what a user must enter in conjunction with the user’s password when logging in to MicroStrategy.

• The user name is the name displayed and associated with a user login.

• The group name is the name displayed and associated with a group.

To select the LDAP attributes to import for user and group information

1 In MicroStrategy Desktop, 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 User/Group. The User/Group Import options are displayed.
4 In the **Import user login as** area, select one of the following LDAP attributes to import as the user login:

- **User login**: Imports the user’s LDAP user login as the MicroStrategy user login.
- **Distinguished name**: Imports the user’s LDAP Distinguished Name as the MicroStrategy user login.
- **Other**: You can provide a different LDAP attribute than those listed above to be imported and used as the MicroStrategy user login. Your LDAP administrator can provide you with the appropriate LDAP attribute to be used as the user login.

If you type a value in the **Other** field, ensure that the authentication user contains a valid value for the same attribute specified in the **Other** field. For more information on the authentication user, see *Authentication user, page 134*.

5 In the **Import user name as** area, select one of the following LDAP attributes to import as the user name:

- **User name**: Imports the user’s LDAP user name as the MicroStrategy user name.
- **Distinguished name**: Imports the user’s LDAP Distinguished Name as the MicroStrategy user name.
- **Other**: You can provide a different LDAP attribute than those listed above to be imported and used as the MicroStrategy user name. Your LDAP administrator can provide you with the appropriate LDAP attribute to be used as the user name.

If you type a value in the **Other** field, ensure that the authentication user contains a valid value for the same attribute specified in the **Other** field. For more information on the authentication user, see *Authentication user, page 134*.

6 In the **Import group name as** area, select one of the following LDAP attributes to import as the group name:

- **Group name**: Imports the group’s LDAP group name as the MicroStrategy group name.
- **Distinguished name**: Imports the group’s LDAP Distinguished Name as the MicroStrategy group name.
Other: You can provide a different LDAP attribute than those listed above to be imported and used as the MicroStrategy group name.

Your LDAP administrator can provide you with the appropriate LDAP attribute to be used as the group name.

If you type a value in the Other field, ensure that the authentication user contains a valid value for the same attribute specified in the Other field. For more information on the authentication user, see Authentication user, page 134.

Importing integrated authentication credentials associated with LDAP users

If you are importing users in a batch using an LDAP-based single sign-on system, you can import the single sign-on ID whenever a user logs into the system or is imported into the system. For information about configuring a single sign-on system, see Enabling single sign-on authentication, page 169.

By default the single sign-on information is stored in the userPrincipalName LDAP attribute. If your system stores single sign-on information in a different LDAP attribute, you can specify the attribute when you configure the import.

To import integrated authentication credentials associated with LDAP users

1. In MicroStrategy Desktop, 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 Import Options. The Import options are displayed.

4. Select the Batch import Integrated Authentication/Trusted Authentication unique ID check box.

5. Select whether to use the default LDAP trusted authentication attribute of userPrincipalName, or to use a different attribute. If you want to use a different attribute, specify it in the text field.
Importing email addresses associated with LDAP users

LDAP users often have email addresses associated with them. 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 Scheduling deliveries to email, file, and printer using Distribution Services, page 564.

MicroStrategy 9 imports only the primary email address for each LDAP user.

To import LDAP email addresses associated with LDAP users

1. In MicroStrategy Desktop, 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.

4. Select the Import Email Address check box.

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.

Selecting how many nested groups to import with the user

If your LDAP organizational structure includes groups contained within groups, you can specify how many recursive groups to import when you import a user or group into MicroStrategy.

Controlling how many groups are imported into MicroStrategy is controlled by the Number of Levels to Recurse setting. This setting can be found in the Filters subcategory of the LDAP category of the MicroStrategy
Intelligence Configuration Editor. To understand how this setting effects the way the users and groups are imported into MicroStrategy, see the following table:

If the number 2 is selected for this field, when MicroStrategy imports LDAP groups, it will import the groups associated with each user, 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.

To configure how many levels of groups will be imported

1. In MicroStrategy Desktop, 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, and then select Filters. The Filter options are displayed.

4. In the Number of Groups to Recurse field, select the number of levels of groups that you want to import.
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.

High-level steps to set up links between LDAP users/groups and MicroStrategy users/groups

1. In MicroStrategy Desktop, log in to a project source. You must log in as a user with administrative privileges.

2. In the project source, expand Administration, and then expand User Manager.

3. Depending on whether you want to link a MicroStrategy user or group to an LDAP user or group respectively, perform one of the following:
   - **User**: Expand a MicroStrategy group, and then right-click a user and select Edit. The User Editor opens.
• **Group**: Right-click a group and select **Edit**. The Group Editor opens.

4 Select the **Authentication** tab.

5 In the field within the **LDAP Authentication** area, enter the user distinguished name or the group distinguished name.

6 Click **OK** to accept the changes and close the User Editor or Group Editor.

7 From the **Administration** menu, select **Server**, and then select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens.

8 Expand the **LDAP** category, then expand **Import**, and then select **Import/Synchronize**. The User/Group Import options are displayed.

9 In the **Import/Synchronize at Login** area, clear the **Import Users** and/or **Import Groups** check boxes.

    If the **Import** check boxes are not cleared, the linked MicroStrategy user will be overwritten in the metadata by the imported LDAP user and cannot be recovered.

For specific instructions on accessing and using the User Editor and Intelligence Server Configuration Editor, see the Desktop Help.

### 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 group 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.

**High-level steps to set up anonymous / guest users**

1. Make sure the LDAP server is set up to accept anonymous authentication.

2. In MicroStrategy Desktop, log in to a project source. You must log in as a user with administrative privileges.

3. Right-click the project source that you want the anonymous/guest users to have access to.

4. Select **Modify Project Source**. The Project Source Manager opens.

5. In the Project Source Manager, make sure **Use LDAP Authentication** is selected.

6. Log in to MicroStrategy using any LDAP server user login that is not imported or linked to a MicroStrategy user.

For specific instructions on accessing and using the User Editor and Intelligence Server Configuration Editor, see the Desktop Help.

**Synchronizing imported LDAP users and groups**

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 the Synchronize Users or Synchronize Groups options are selected in the Intelligence Server Configuration Editor, 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. The steps for this process are detailed in To import users and/or groups at login, page 142.

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:**
  - 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 MicroStrategy and both have been imported into MicroStrategy. Then User1 is removed from Group1 in LDAP but Group2 is not imported or linked to a MicroStrategy group. Upon synchronization, User1 is removed from Group1 and is recognized as a member of Group2, but any permissions for Group2 are not applied for the user until Group2 is imported or linked to a MicroStrategy group. In the meantime, User1 is given the security of the LDAP Users group.

- MicroStrategy users and groups that have been imported from the LDAP directory or linked to LDAP users and groups are NOT automatically deleted when their associated LDAP users and groups are deleted from the LDAP directory. 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. The images below show a sample LDAP directory with user Joe Doe being moved within the LDAP directory from Sales to Marketing. Also, the user name for Joe Doe is changed to Joseph Doe, and the group name for Marketing is changed to MarketingLDAP.

The following table describes what happens with users and groups in MicroStrategy if users, groups, or both users and groups are synchronized.

<table>
<thead>
<tr>
<th>Sync Users?</th>
<th>Sync Groups?</th>
<th>User Name After Synchronization</th>
<th>Group Name After Synchronization</th>
<th>User Membership After Synchronization</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>Joe Doe</td>
<td>Marketing</td>
<td>Sales</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Joe Doe</td>
<td>MarketingLDAP</td>
<td>Sales</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Joseph Doe</td>
<td>Marketing</td>
<td>Marketing</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Joseph Doe</td>
<td>MarketingLDAP</td>
<td>MarketingLDAP</td>
</tr>
</tbody>
</table>

In another scenario, User A belongs to Groups 1, 2, and 3 in LDAP. User A is imported into MicroStrategy, along with his groups. Subsequently, User A is added to Group 4 in the LDAP directory. If the Synchronize Users and Synchronize Groups options are selected, the next time User A logs into
MicroStrategy, Group 4 is automatically imported into MicroStrategy metadata and is assigned as an additional group to User A.

If you have trouble getting synchronization to work properly, see *Troubleshooting LDAP authentication, page 767*.

**Synchronization at login**

You can choose to synchronize users and their associated groups when a user logs into MicroStrategy. By synchronizing users and groups at login, a user's user name, group name, and group access are updated every time a user logs into MicroStrategy.

**To synchronize users and/or groups at login**

1. In MicroStrategy Desktop, 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 *Import/Synchronize*. The Import/Synchronize options are displayed.

4. In the *Import/Synchronize at Login* area select the Synchronize check boxes for users and groups, depending on whether you want to synchronize users, groups, or both at login.

5. Click OK to accept your changes and close the Intelligence Server Configuration Editor.

**Batch synchronization**

Rather than synchronizing users and their associated groups every time a user logs in to MicroStrategy, you can choose to synchronize users and groups in batch. By synchronizing users and groups in batch, users and groups are synchronized on the schedules your select (see *Synchronization schedules, page 158*).
To synchronize users and/or groups in batch

1 In MicroStrategy Desktop, 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 Import/Synchronize. The Import/Synchronize options are displayed.

4 In the Import/Synchronize in Batch area select the Synchronize check boxes for users and groups, depending on whether you want to synchronize users, groups, or both in batch.

5 Depending on whether you want to synchronize users, groups, or both, enter a search filter to return a list of users or groups to synchronize in one of the following fields:

   • Enter search filter for importing list of users: Enter a search filter that is used to synchronize a list of users in batch. For more information on defining a user search filter, see Defining a search filter to return a list of users, page 145.

   • Enter search filter for importing list of groups: Enter a search filter that is used to synchronize a list of groups in batch. For more information on defining a group search filter, see Defining a search filter to return a list of groups, page 146.

6 You can setup MicroStrategy to synchronize users and groups on schedules. For more information, see the following section Synchronization schedules, page 158.

7 Click OK to accept your changes and close the Intelligence Server Configuration Editor.

Synchronization schedules

If you choose to synchronize users and groups in batch, you can select a schedule that dictates when LDAP users and groups are synchronized in MicroStrategy. For more information on creating and using schedules, see Creating and managing schedules, page 547.
To synchronize users and groups on schedules

1. In MicroStrategy Desktop, 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, and then select Schedules. The Schedules options are displayed.

4. In the Synchronization Schedules area select the check boxes next to the schedules you want to use to synchronize users and groups.

   Note the following:
   
   - Users and groups are only synchronized with the selected schedules if the synchronize users and synchronize groups in batch options are selected respectively. For steps to synchronize users and/or groups, see To synchronize users and/or groups in batch, page 158

   - To run the selected schedules after accepting your changes and exiting the MicroStrategy Intelligences Server Configuration Editor, select the Run schedules now check box.

5. Click OK to close the Intelligence Server Configuration Editor.

Managing LDAP authentication in MicroStrategy

The sections above detail different concepts, procedures, and suggestions that can get you started with implementing LDAP with your MicroStrategy environment. While working with MicroStrategy and implementing LDAP authentication, you may come across situations that could benefit from better performance or require troubleshooting. The sections below cover concepts that can help your LDAP authentication and MicroStrategy systems work as a cohesive unit.
Implementing different LDAP security types

You can tune your security setup using the following options:

- Using authentication binding or password comparison, page 160
- Enabling database passthrough authentication with LDAP, page 161
- Linking a Windows login to an LDAP user, page 162

These options are explained below, with steps to configure them.

Using 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 LDAP user by 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.

To enable binding or password comparison authentication

Performing this procedure may compromise one or more security features in your environment. Enabling password comparison only configures MicroStrategy to authenticate LDAP users by only
verifying that the user typed in the correct password. No other verifications are performed.

1 In MicroStrategy Desktop, 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, and then select Server. The Server options are displayed.

4 Under Authentication method select one of the following:
   - **Binding**: When an LDAP user logs into MicroStrategy, their user name, password, and other user account restrictions and statuses are verified to authenticate the user.
   - **Password comparison**: When an LDAP user logs into MicroStrategy, only their user name and password are verified to authenticate the user.

On the LDAP server, make sure that the password comparison functionality is enabled. If this functionality is disabled on the LDAP server, MicroStrategy uses binding automatically.

### Enabling 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 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 62.

Scheduled tasks may fail to run when using database passthrough authentication if a user’s password is changed during a session in MicroStrategy. 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.

---

**To support database passthrough authentication**

1. In MicroStrategy Desktop, log in to a project source. You must log in as a user with administrative privileges.

2. In Desktop’s Folder List, expand **Administration**, and then expand **User Manager**.

3. Expand a MicroStrategy group, then right-click a user and select **Edit**. The User Editor opens.

4. Select the **Authentication** tab.

5. In the **Warehouse Passthrough** area, enter the following information:
   - **Warehouse passthrough login**: Enter the LDAP user login to connect to and execute reports or queries against the database.
   - **Warehouse passthrough password**: Enter the password for the LDAP user login.

6. Click **OK** to accept your changes and close the User Editor.

**Linking a Windows login to an LDAP user**

When using LDAP authentication in 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.

For example, a user logs in to their Windows machine with a linked LDAP login and password and is authenticated. The user then opens MicroStrategy Desktop 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 MicroStrategy Desktop, 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.

6. You must also define your MicroStrategy system to authenticate users using Windows authentication. For information on implementing this authentication, see Implementing Windows NT authentication, page 169.
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.

By not using connection pooling, the connection to an LDAP server is closed after each request, which can help reduce the use of network resources if requests are sent to the LDAP server infrequently.

To enable connection pooling

1. In MicroStrategy Desktop, 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, and then select Server. The Server options are displayed.

4. Select the check box Use connection pooling to enable the reuse of open connections to the LDAP server. Clearing this check box causes MicroStrategy to open and close a connection to the LDAP server for every operation.

5. Click OK to accept your changes and close the Intelligence Server Configuration Editor.

Connection pooling with clustered LDAP servers

To implement LDAP you may have multiple LDAP servers which work together as a cluster of LDAP servers.

When a request to open an LDAP connection is made, the LDAP server with the least amount of 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.
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 107*. For general information about security filters, see *Restricting access to data: Security filters, page 95*. 
To import an LDAP attribute into a project

1. In Desktop, 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.

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 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 Desktop, 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 login fails if LDAP attribute value is not read from the LDAP server** check box.
5. Click **OK**. You may need to restart Intelligence Server for this change to take effect.

**Troubleshooting**

While the sections above on implementing LDAP authentication in MicroStrategy are intended to facilitate the use of LDAP authentication in MicroStrategy successfully, there are still situations where you can come across problems or errors while trying to integrate the two systems. For troubleshooting concepts, information, and procedures, see *Troubleshooting LDAP authentication, page 767*. 

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 enable single sign-on in MicroStrategy. These types of authentication are:

- Windows — Implementing Windows NT authentication, page 169
- Integrated - Enabling integrated authentication, page 176
- Tivoli or SiteMinder - Enabling Single Sign-on to Web, Mobile, and Office with third-party authentication, page 199

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 171.

To use Windows authentication you must create users in the MicroStrategy environment and then link them to Windows users. Linking enables MicroStrategy Intelligence Server to map a Windows user to a MicroStrategy user. See Linking a Windows domain user to a MicroStrategy user, page 173.

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 Desktop Help.

This type of authentication relies heavily on Microsoft technologies. In particular, Intelligence Server must be running in a Windows environment. If Intelligence Server is running in a UNIX environment, Windows authentication is not supported in Desktop. However, Windows users can be
imported from Desktop when Intelligence Server is running in a UNIX environment.

To use Windows authentication with MicroStrategy Web, you must be running Web or Web Universal under Microsoft IIS. Non-IIS web servers do not support Windows authentication. See Enabling integrated authentication for IIS, page 184

Beginning with MicroStrategy 8.1.0, Windows authentication can be used with MicroStrategy Web on IIS regardless of the operating system on which Intelligence Server is installed.

If the Windows domain account information is linked to a MicroStrategy user definition, a MicroStrategy Web user can be logged in automatically through Web. When a user accesses MicroStrategy Web, IIS detects the Windows user and sends the login information to MicroStrategy 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 174.

**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.

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 171.

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 Web Server 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


2. 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 173.

3. Define a project source to use Windows authentication. See Defining a project source to use Windows authentication, page 173.


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.

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 5**: See [http://support.microsoft.com/kb/215383](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.
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 MicroStrategy Desktop, 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.

   If you are using MicroStrategy 8.1.1, use a project source with a direct (two-tier) connection to map users.

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. On the Authentication tab, 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.

5. Click OK to save your changes and close the User 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 MicroStrategy Desktop, 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.

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 174.

For steps to enable Windows authentication for a project, see To enable Windows authentication login for a project, page 175.

To enable Windows authentication login for MicroStrategy Web

1. From the Windows Start menu, point to Programs, then point to MicroStrategy, then point to Web, 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 top of the page, click **Preferences**.

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**.

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 MicroStrategy Desktop or MicroStrategy Web. This type of authentication uses Kerberos to validate a user's credentials.

In addition to authenticating users to MicroStrategy Desktop 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.
<table>
<thead>
<tr>
<th>Machine</th>
<th>Required Configurations</th>
</tr>
</thead>
</table>
| 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 178**.  
• If Intelligence Server is run as an application with a particular user account, you 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 179**.  
• 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 179**.  
• Define the web server for MicroStrategy Web host machine to be trusted for delegation. This is achievable by selecting the **Trust computer for delegation** authentication option for the host machine. For information on this configuration, see **Trusting the MicroStrategy Web server host for delegation, page 179**. |
| 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 180**. |
| 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 184**.  
To enable single sign-on authentication to MicroStrategy Web 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 186**.  
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 187**. |
| Machine hosting a J2EE-compliant application server | If you use an application server other than IIS to deploy MicroStrategy Web Universal 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 188**. |
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.

### Machine | Required Configurations
--- | ---
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 195](#).
Any machine with the required software for the task | In MicroStrategy Desktop, link a MicroStrategy user to the domain user. For information on this configuration, see [Linking a domain user to a MicroStrategy user, page 195](#).
Any machine with the required software for the task | In MicroStrategy Desktop, configure a project source to use integrated authentication. For information on this configuration, see [Using integrated authentication for a project source, page 196](#).
Any machine with the required software for the task | 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 197](#).
Any machine with the required software for the task | In addition to authenticating users to MicroStrategy Desktop 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 197](#).
**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:

- Create a Service Principal Name (SPN) for the Intelligence Server. The SPN must be in the following format: `MSTRSVRSvc/ISMachineName:ISPort@DOMAIN_REALM`, where `ISMachineName` is the fully qualified host name of the machine, `ISPort` is the port number defined for the Intelligence Server, and `DOMAIN_REALM` is the domain realm for which you are setting up integrated authentication.

  If you are running Intelligence Server as an application, `DOMAIN_REALM` is required. If you are running Intelligence Server as a service, `DOMAIN_REALM` is optional.

  An example of a valid SPN is: `MSTRSVRSvc/iserver-host:34952@EXAMPLE.COM`

- Trust Intelligence Server for delegation. For the user account that Intelligence Server runs under, enable the **Account is trusted for delegation** 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 also configure the Intelligence Server machine, as described in Configuring Intelligence Server Universal on UNIX/Linux for integrated authentication, page 180.

**Trusting the MicroStrategy Web server host for delegation**

The web server host for MicroStrategy Web 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 server machine in your domain controller. You must select the **Trust this**
computer for delegation to any service (Kerberos only) option for the MicroStrategy Web server machine.

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.

The configurations listed below are required to configure Intelligence Server Universal with your Windows domain controller and Kerberos security:

- *Install Kerberos 5, page 180*
- *Ensure that the environment variables are set, page 181*
- *Create a Service Principal Name for the Intelligence Server, page 181*
- *Configure the krb5.keytab file for the Intelligence Server, page 182*
- *Configure the krb5.conf file for the Intelligence Server, page 183*

Kerberos only supports US-ASCII characters. Do not use any special characters when installing or configuring Kerberos.

**Prerequisites**

- 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:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Default</th>
<th>Required/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(KRB5_HOME)</td>
<td>Location of all Kerberos configuration files</td>
<td>/etc/krb5</td>
<td>Optional</td>
</tr>
<tr>
<td>$(KRB5_CONFIG)</td>
<td>Location of the default Kerberos configuration file</td>
<td>/etc/krb5/krb5.conf</td>
<td>Required</td>
</tr>
<tr>
<td>$(KRB5_CCNAME)</td>
<td>Location of the Kerberos credential cache</td>
<td>/etc/krb5/krb5_ccache</td>
<td>Optional</td>
</tr>
<tr>
<td>$(KRB5_KTNAME)</td>
<td>Location of the Kerberos keytab file</td>
<td>/etc/krb5/krb5.keytab</td>
<td>Required</td>
</tr>
</tbody>
</table>

Create a Service Principal Name for the Intelligence Server

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.

**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/ISMACHINE_NAME: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/ISMACHINE_NAME: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/ISMACHINE_NAME:ISPort@DOMAIN_REALM

   The command should run without prompting you for a username and password.

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 Microsoft Analysis Services. 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 5:** [http://support.microsoft.com/kb/215383](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 197*.

If you use a J2EE-compliant application server other than IIS to deploy MicroStrategy Web Universal or MicroStrategy Web Services, see *Enabling integrated authentication for J2EE-compliant application servers, page 188*.

---

**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*.


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 Web Services as well, repeat the above procedure for the **MicroStrategyWS** virtual folder.
9 Restart IIS for the changes to take effect.

**Creating a Service Principal Name for IIS**

You must 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, you must 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 listed below is produced by third-party vendors 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](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\Common Files\MicroStrategy\                  

The path listed above assumes you have installed MicroStrategy in the C:\Program Files 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, 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 188
• Configure the krb5.keytab file for the application server, page 189
• Configure the krb5.conf file for the application server, page 191
• Configure the jaas.conf file for the application server, page 192
• Configure the JVM startup parameters, page 194
• Enable the SPNEGO mechanism, page 194

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 190*.

The steps to configure this file on a Windows machine are provided in *To create a krb5.keytab file in Windows, page 191*.

**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:

   ```shell
   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:

   ```shell
   ktutil
   addent -password -p HTTP/ASMachineName@DOMAIN_NAME -k KeyVersionNumber -e EncryptionType
   wkt /etc/krb5/krb5.keytab
   exit
   ```
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 189*, 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**

```java
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**

```java
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**

```java
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 deployment, you must modify MicroStrategy’s `web.xml` file 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:

```
<filter>
    <display-name>SpnegoFilter</display-name>
    <filter-name>SpnegoFilter</filter-name>
    <filter-class>com.microstrategy.web.filter.SpnegoFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>SpnegoFilter</filter-name>
    <servlet-name>mstrWeb</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 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 178`.

- 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 MicroStrategy Desktop, 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 On the **Authentication** tab, in the **Trusted Authentication Request / Integrated Authentication** area, type the domain user in the **User ID** field. Valid syntax is shown below:

   \[\text{DomainUserName@DOMAIN_REALM}\]

   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.

5 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.

**To use integrated authentication for a project source**

1 In MicroStrategy Desktop, 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 *Create a Service Principal Name for the Intelligence Server, page 181*. For example, if the SPN is `MSTRVRSvc\server.example.com:1234`, the Server Name for the project source should be `server.example.com`. 
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 **Programs**, then **MicroStrategy**, then **Web**, 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 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.

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 Desktop, 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.

8. Click OK. The Project Configuration dialog box closes.
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 Tivoli, SiteMinder, or Oblix:

- MicroStrategy Web
- MicroStrategy Mobile (Tivoli and SiteMinder only)
- MicroStrategy Web Services, to support MicroStrategy Office (Tivoli and 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 and Mobile. 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 are URL-decoded by default within MicroStrategy Web or Mobile 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, and each is detailed below:

- *Creating users and links in Tivoli, SiteMinder, or Oblix, page 201*
- *Enabling single sign-on authentication to MicroStrategy Web, Mobile or Office, page 202*
• Importing and linking Tivoli/SiteMinder users in MicroStrategy, page 211

• To log in to MicroStrategy Web using Tivoli single sign-on, page 216

Creating users and links in Tivoli, SiteMinder, or Oblix

Before MicroStrategy can be configured to accept Tivoli, SiteMinder, or Oblix authentication, certain preliminary settings must be established. This ensures that a link exists between Tivoli or SiteMinder and MicroStrategy Web, and that the link is functioning as required.

You must complete all of the following steps to ensure proper configuration of Tivoli or SiteMinder and MicroStrategy products.

Creating a Tivoli, SiteMinder, or Oblix user

You can enable SSO authentication in MicroStrategy Web by associating a MicroStrategy user to a user in Tivoli or SiteMinder. To test this association, you must create a Tivoli or SiteMinder user to confirm that access has been properly configured in MicroStrategy products.

For steps to create a new user, refer to your Tivoli or SiteMinder documentation.

Creating a Tivoli, SiteMinder, or Oblix link to MicroStrategy applications

You link Tivoli to MicroStrategy applications using junctions, SiteMinder using Web Agents, and Oblix using Webgates. These links redirects users from the 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.

Oblix authentication is only available for MicroStrategy Web.

For steps to create a junction (in Tivoli) a Web Agent (in SiteMinder), or a Webgate (Oblix), 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 first establish trust between Office and the Intelligence server, and then enable trusted authentication in the configuration files for Web Services.

This section explains the following required steps to enable SSO authentication in MicroStrategy Web or Mobile:

• Enabling trusted authentication in MicroStrategy Web, page 202
• Enabling trusted authentication in MicroStrategy Mobile, page 204
• Establishing trust between MicroStrategy Web or Mobile and Intelligence Server, page 204
• Establishing trust between MicroStrategy Web Services and Intelligence Server, to support MicroStrategy Office, page 208
• Enabling trusted authentication in MicroStrategy Web Services to support MicroStrategy Office, page 209

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. This is discussed in Enabling anonymous authentication for Internet Information Services, page 210.

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.
To enable trusted authentication in MicroStrategy Web

1. From the Windows Start menu, point to Programs, then MicroStrategy, then Web, 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:

<table>
<thead>
<tr>
<th>Login mode</th>
<th>Enabled</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard (user name &amp; password)</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>LDAP Authentication</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Database Authentication</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Guest</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Windows Authentication</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Integrated Authentication</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Trusted Authentication Request</td>
<td>✔️</td>
<td>○</td>
</tr>
</tbody>
</table>

4. From the Trusted Authentication Providers drop-down list, select Tivoli, SiteMinder, or Oblix.

   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

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.

**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 Programs, then MicroStrategy, then Mobile, 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 Tivoli or SiteMinder.

   For information about adding custom authentication, see the MicroStrategy SDK documentation.

4. At the bottom of the page, click Save.

To create a mobile configuration to send to users’ iPhones or iPads, 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 Tivoli 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 207.*

If you are using multiple Intelligence Server machines in a cluster, you must first set up the cluster, as described in *Chapter 8, Clustering Multiple MicroStrategy Servers,* and then establish trust between Web or Mobile Server and the cluster.

---

**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 **Programs**, then **MicroStrategy**, then **Web**, and then select **Web Administrator**. The MicroStrategy Web Administrator page opens.
   - From the Windows **Start** menu, point to **Programs**, then **MicroStrategy**, then **Mobile**, 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.

<table>
<thead>
<tr>
<th>Server Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection properties</strong></td>
</tr>
<tr>
<td><strong>Server name</strong></td>
</tr>
<tr>
<td><strong>Connected</strong></td>
</tr>
<tr>
<td><strong>Connect mode</strong></td>
</tr>
<tr>
<td><strong>Port</strong></td>
</tr>
<tr>
<td><strong>Initial pool size</strong></td>
</tr>
<tr>
<td><strong>Maximum pool size</strong></td>
</tr>
<tr>
<td><strong>Load balance factor</strong></td>
</tr>
</tbody>
</table>

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.

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 the URL for MicroStrategy Web or Mobile, as applicable. For example, you can provide 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 Programs, then MicroStrategy, then Desktop, and then select Desktop. MicroStrategy Desktop 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 SSO category, and verify that the trusted relationship is listed in the Trusted Web Application Registration list.

To delete a trust relationship

1 Open MicroStrategy Web Administrator or MicroStrategy Mobile Administrator, as applicable:

- From the Windows Start menu, point to Programs, then MicroStrategy, then Web, and then select Web Administrator. The MicroStrategy Web Administrator page opens.

- From the Windows Start menu, point to Programs, then MicroStrategy, then Mobile, and then select Mobile Administrator. The MicroStrategy Mobile Server Administrator page opens.

2 On the left, click Servers.

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 or Trust relationship between MicroStrategy Mobile Server and MicroStrategy Intelligence Server, as applicable, click Delete.
The Delete Trust with MicroStrategy Intelligence Server page opens.

6 Provide your login information in the appropriate fields.

7 Click **Delete trust relationship**.

![Delete Trust Relationship with MicroStrategy Intelligence Server](image)

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 connect to the project source you want to use trusted authentication for, and then establish the trust relationship between Office and the Intelligence Server. Once you have completed this step, you must edit the `projectsources.xml` file for Web Services. 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 MicroStrategy Office toolbar, click **MicroStrategy Office**. MicroStrategy Office starts, with a list of project sources you can connect to.

3 In the left pane, 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.

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 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 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

1 On the machine where MicroStrategy Web Services is installed, open the `projectsources.xml` file. By default, the file is located in `C:\Program Files\MicroStrategy\Web Services`.

2 In the `projectsources.xml` file, locate the `<ProjectSource>` tag describing the project source you want to enable SSO for.

3 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>
```

4 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 5**: [http://support.microsoft.com/kb/310344](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.


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.


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 Tivoli/SiteMinder users in MicroStrategy

For Tivoli or SiteMinder users to access MicroStrategy applications, the users must be granted MicroStrategy privileges. The following flow chart
illustrates the various ways that MicroStrategy users are handled when they log in to Tivoli or SiteMinder.

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 213.

- 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 215.

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 214.

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.

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 Programs, then MicroStrategy, then Desktop, and then select Desktop. MicroStrategy Desktop opens.

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 SSO category.

5. On the right, select the Import user at login check box.

6. Click OK to save your changes and exit 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 Programs, then MicroStrategy, then Desktop, and then select Desktop. MicroStrategy Desktop 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.


7. In the Trusted Authentication Request field, type the Tivoli user name to link to the MicroStrategy user.

   The name you type in the Trusted Authentication Request 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.

To enable guest access to MicroStrategy Web or Mobile for Tivoli users

1. From the Windows Start menu, point to Programs, then MicroStrategy, then Desktop, and then select Desktop. MicroStrategy Desktop 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 SSO** 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 SSO-MicroStrategy user link can not be 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 save 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:
   ```url
   https://MachineName/JunctionName/MicroStrategyWebURL
   ```
   where:
   - **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.

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.

### 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 110.
To establish this configuration

1. In Desktop, 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 the User Editor, select the **Authentication** tab, and link users to their respective database user IDs using the **Warehouse Login** and **Password** boxes for each user. For details on each option, click **Help**.

4. 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.

For example, perhaps you are partitioning fact tables by rows, as described in *Splitting fact tables by rows, page 110*. 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 Desktop, 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.

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.
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 222*
- *Auditing and updating licenses, page 225*
- *Updating CPU affinity, page 230*
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:

- **Named User licenses**, page 222, in which the number of users with access to specific functionality are restricted
- **CPU licenses**, page 224, 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 226.

When you obtain additional licenses from MicroStrategy, use License Manager to update your license information. For details, see *Updating your license*, page 229.

**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 226*).

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.

**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 Desktop privilege in the Desktop Analyst group are granted Desktop Analyst or Desktop Designer licenses. Users who do not have the Use Desktop 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 225*.

For steps to manually verify your Named User licenses using License Manager, see *Auditing your system for the proper licenses, page 228*.

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 Desktop, 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)**.

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 226.*

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 230.*

---

### 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 225.*

For steps to manually verify your CPU licenses using License Manager, see *Auditing your system for the proper licenses, page 228.*
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 Desktop. 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 223 and Verifying CPU licenses, page 224.

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.
• 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 MicroStrategy 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.

• 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 Start menu, point to Programs, then MicroStrategy, 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. Type mlicmgr 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 the size of your metadata is too large for the Java Virtual Machine heap size. For steps to modify the Java Virtual Machine heap size in 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 MicroStrategy Intelligence Server. You can use either MicroStrategy Configuration Wizard or MicroStrategy Desktop'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 227*.

   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 *MicroStrategy 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 227*.

   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.

### 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 Start menu, point to MicroStrategy, then 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 MicroStrategy 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.

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 greyed 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 234.

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.

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 MicroStrategy 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.

**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. 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\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.

**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.
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 240
- *Implementing the recommended life cycle*, page 244
- *Duplicating a project*, page 246
- *Updating projects with new objects*, page 253
- *Copying objects between projects: Object Manager*, page 256
- *Merging projects to synchronize objects*, page 296
- *Comparing and tracking projects*, page 304
- *Deleting unused schema objects: managed objects*, page 309

For information about creating a project, creating attributes and facts, building a logical data model, and other project design tasks, see the *MicroStrategy 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 241
- For a real-life scenario, see *Real-life scenario: New version from a project developer*, page 243
- For details on how to implement the project life cycle in your MicroStrategy environment, see *Implementing the recommended life cycle*, page 244
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 246.

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 MicroStrategy 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 using the Project Duplication Wizard. For detailed information about the Project Duplication Wizard, see Duplicating a project, page 246.

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 256.

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,
For detailed information about Project Merge, see *Merging projects to synchronize objects*, page 296.

To help you decide whether you should use Object Manager or Project merge, see *Comparing Project Merge to Object Manager*, page 254.

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 304. For instructions on how to use search objects to track changes in a project, see *Tracking your projects with the Search Export feature*, page 307.

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 13, 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 300.
This combination of the two projects creates Project version 2.1, as shown in the diagram below.

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 296.

**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 246*.

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 241*.

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 253*.

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 254*. For a tool to determine what objects have changed, see *Comparing and tracking projects, page 304*. 

© 2011 MicroStrategy, Inc.

Implementing the recommended life cycle 245
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 241.

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 13, 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 253.

7 Repeat steps 3 through 6 as necessary.

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 249.

To migrate a project from one database platform to another, you must use the Project Mover Wizard. For detailed information about this migration, see Migrating a project to a new database platform, page 251.

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 a MicroStrategy 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 Chapter 4 of the MicroStrategy 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 MicroStrategy 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 268*.

### 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 the MicroStrategy Object Manager select the **Project** menu (or from MicroStrategy Desktop 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\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 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\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 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.

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

• 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 MicroStrategy Upgrade Guide.

**To migrate a project to a different database**

1. From the **Start** menu point to **Programs**, then **MicroStrategy**, then **Tools**, and then select **Project Mover**. The Project Mover Wizard opens.

2. Select the 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\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 to the same specifications, but there is always the possibility that you could 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 256*.

- 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 276*.

- **Project Merge** migrates all the objects in a project at once. For information about Project Merge, see *Merging projects to synchronize objects, page 296*. 
For a comparison of these tools, see *Comparing Project Merge to Object Manager, page 254.*

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 246.*

- 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 276.

### 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 257.

A metadata lock prevents other MicroStrategy users from modifying any objects in the project in MicroStrategy Desktop 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 MicroStrategy Desktop 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 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 MicroStrategy Desktop. For detailed steps on locking and unlocking projects manually, see the Desktop Help (press F1 from within Desktop).

Command Manager scripts can be used to automate metadata lock management. For information about Command Manager, see Chapter 12, 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 254.

This section includes:

- Prerequisites for copying objects between projects, page 256
- Project locking with Object Manager, page 257
- Copying objects, page 258
- What happens when you copy or move an object, page 262
- Resolving conflicts when copying objects, page 269
- Copying objects in a batch: Update packages, page 276

Prerequisites for copying objects between projects

- To use Object Manager to copy objects between projects, you must have the Use Object Manager privilege. 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 Checks privilege when that user is using Object Manager.

- 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 246.

- 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 *MicroStrategy 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 may be able to move objects from the older version to the updated project.

**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 255.

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 child 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 276*.

**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.

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 277*.

For background information on these objects, including how they are created and what roles they perform in a project, see the *MicroStrategy 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 265*. For details about the Conflict Resolution options, see *Resolving conflicts when copying objects, page 269*. For details about the Migration options, see *What happens when you copy or move an object, page 262*. 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 276.

• 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.

• 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 268.

• 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 13, 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 246*.

**Log in to the projects in Object Manager**

1. From the Windows *Start* menu, point to *Programs*, then *MicroStrategy*, then *Object Manager*, 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.

   Use the appropriate sub-procedure below depending on whether you want to *Copy application and schema objects* or *Copy configuration objects*.

**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 Child dependencies, page 263.

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 269.

For more information about handling specific situations, see:

- Managing object dependencies, page 263
- Migrating dependent objects, page 265
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: child dependencies and parent dependencies.

When you migrate an object to another project, by default any objects used by that object in its definition (its child 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 267*.

Child dependencies

A child 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 child dependency on the Revenue base formula. (Additionally, the Revenue base formula has a parent dependency of the Revenue metric.)

When you migrate an object to another project, any objects used by that object in its definition (its child dependencies) are also migrated.

To manage child or parent 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 child dependencies. The Child dependencies dialog box opens and displays a list of objects that the selected object uses in its definition. The image below shows the child
dependencies of the Revenue metric in the MicroStrategy Tutorial project: in this case, the child dependency is the Revenue base formula.

3 In the Child dependencies dialog box, you can do any of the following:

- View child dependencies for any object in the list by selecting the object and clicking the Object child dependencies toolbar icon.

- Open the Parent dependencies dialog box for any object in the list by selecting the object and clicking the Object parent dependencies icon on the toolbar. For information about parent dependencies, see Parent dependencies, page 264.

- 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.

Parent dependencies

A parent 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 parent dependencies of many reports and even other metrics. The Revenue metric is said to be a child of these other objects.

Parent dependents are not automatically migrated with their children. However, you cannot delete an object that has parent dependencies without first deleting its children.

To manage the parent 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 parent dependencies. The Parent 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 parent dependencies for the Revenue metric in the MicroStrategy Tutorial project.

![Parent dependencies of Revenue](image)

3 In the Parent dependencies dialog box, you can do any of the following:

- View parent dependencies for any object in the list by selecting the object and clicking the **Object parent dependencies** icon on the toolbar.

- Open the Child dependencies dialog box for any object in the list by selecting the object and clicking the **Object child dependencies** icon on the toolbar. For information about child dependencies, see **Child dependencies, page 263**.

- 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 child 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 Dependencies to make sure all child 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 child 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 child dependency on its contents.

- **Shortcut objects** have a child 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 child 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.

  Groups have a parent 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 parent 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 child 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 an attribute’s parent attributes** and **Exclude an attribute’s child attributes**: An attribute has a child 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 selected tables of attributes** and **Exclude selected tables of facts**: An attribute or fact has a child 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.

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 Desktop or from the machine’s regional settings.

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 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:

<table>
<thead>
<tr>
<th>Conflict</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists identically</td>
<td>The object ID, object version, and path are the same in the source and destination projects.</td>
</tr>
<tr>
<td>Exists differently</td>
<td>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.</td>
</tr>
<tr>
<td>Exists identically except for path</td>
<td>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. <strong>Note</strong>: 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 <strong>Replace</strong> action, the destination object is updated to reflect the path of the source object.</td>
</tr>
<tr>
<td>Exists identically except for Distribution Services objects</td>
<td>(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 <strong>Replace</strong> action, the destination user is updated to reflect the contacts and contact groups of the source user.</td>
</tr>
<tr>
<td>Does not exist</td>
<td>The object exists in the source project but not in the destination project. <strong>Note</strong>: If you clear the <strong>Show new objects that exist only in the source</strong> 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.</td>
</tr>
</tbody>
</table>

**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 272.*

<table>
<thead>
<tr>
<th>User Action</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing</td>
<td>No change is made to the destination object. The source object is not copied.</td>
</tr>
<tr>
<td>Replace</td>
<td>The destination object is replaced with the source object.</td>
</tr>
<tr>
<td>Note:</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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 child objects are added to it, thus changing the version ID and modification times during the copy process.</td>
</tr>
<tr>
<td>Keep both</td>
<td>No change is made to the destination object. The source object is duplicated in the destination location.</td>
</tr>
<tr>
<td>Use newer</td>
<td>If the source object’s modification time is more recent than the destination object’s, the Replace action is used.</td>
</tr>
<tr>
<td></td>
<td>Otherwise, the Use existing action is used.</td>
</tr>
<tr>
<td>Use older</td>
<td>If the source object’s modification time is more recent than the destination object’s, the Use existing action is used.</td>
</tr>
<tr>
<td></td>
<td>Otherwise, the Replace action is used.</td>
</tr>
<tr>
<td>Merge (user/group only)</td>
<td>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.</td>
</tr>
<tr>
<td>Do not move (table only)</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 274.*
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 *MicroStrategy 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 *MicroStrategy 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 258. For an explanation of each object action, see *Choosing an action to resolve a conflict*, page 270.

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.

---

**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 258
   - For an explanation of each object action, see *Choosing an action to resolve a conflict*, page 270.

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 theTools 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.
   - To use the ACL of the replaced destination object, select Replace
     existing ACL when replacing objects.

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 265.*

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.

<table>
<thead>
<tr>
<th>Object Action</th>
<th>What happens in the destination project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Existing</td>
<td>The object does not reference the table.</td>
</tr>
<tr>
<td>Replace</td>
<td>The object has the same references to the table as it does in the source project.</td>
</tr>
<tr>
<td>Keep Both</td>
<td>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.</td>
</tr>
</tbody>
</table>
• **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.

<table>
<thead>
<tr>
<th>Object Action</th>
<th>What happens in the destination project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Existing</td>
<td>The object does not reference the table.</td>
</tr>
<tr>
<td>Replace</td>
<td>The object has the same references to the table as it does in the source project.</td>
</tr>
<tr>
<td>Keep Both</td>
<td>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.</td>
</tr>
</tbody>
</table>

• **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.

<table>
<thead>
<tr>
<th>Object Action</th>
<th>What happens in the destination project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Existing</td>
<td>The object has the same references to the table as it did before the action.</td>
</tr>
<tr>
<td>Replace</td>
<td>The object does not reference the table.</td>
</tr>
<tr>
<td>Keep Both</td>
<td>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.</td>
</tr>
</tbody>
</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...
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 269.*

In addition to the standard Object Manager conflict resolution rules (see *Choosing an action to resolve a conflict, page 270*), 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 parent 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 295.*
### 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 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 282.

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 284.*

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 282.*

- By default, users cannot create project update packages in read-only mode. This is because objects, and their child 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 257.*

---

**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.

![Create Package dialog box](image)

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.

![Conflict Resolution dialog box](image)

**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.
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.

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**

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 270.*

Select the schema update options for this package. For more details on these options, see *Update packages and updating the project schema, page 294.*

Select the ACL options for objects in this package. For more details on these options, see *Conflict resolution and access control lists, page 273.*

**Saving the package**

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.

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** or **Save As Excel.**

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 278.

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 check box. For information about project security packages, see Updating project access information for users and groups, page 278.

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 270.

   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 273.
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 or Save As Excel.

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:

<table>
<thead>
<tr>
<th>If the XML file specifies a...</th>
<th>The update package contains...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folder</td>
<td>All objects in the folder</td>
</tr>
<tr>
<td>Search object</td>
<td>All objects returned by the search</td>
</tr>
<tr>
<td>Object prompt</td>
<td>All objects returned by the prompt</td>
</tr>
</tbody>
</table>

To create an XML file for a configuration update package, see Manually creating an update package creation XML file, page 287. 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.

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 270.

10 Select the schema update options for this package. For more details on these options, see Update packages and updating the project schema, page 294.

11 Select the ACL options for objects in this package. For more details on these options, see Conflict resolution and access control lists, page 273.

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 255.

Call the Project Merge executable, projectmerge.exe, with the following parameters:

<table>
<thead>
<tr>
<th>Effect</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use this XML file to create an update package (required)</td>
<td>-f Filename.xml</td>
</tr>
<tr>
<td>Log into the project source with this password (the login ID to be used is stored in the XML file)</td>
<td>-sp Password</td>
</tr>
</tbody>
</table>
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 can be found in the Object Manager folder. By default this folder is `C:\Program Files\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 296*.

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.

<table>
<thead>
<tr>
<th>Effect</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log into the project with this password (the login ID to be used is</td>
<td>-smpPassword</td>
</tr>
<tr>
<td>stored in the XML file)</td>
<td></td>
</tr>
<tr>
<td>Suppress status updates (useful for creating an update package</td>
<td>-sup</td>
</tr>
<tr>
<td>in the background, so that the status window does not appear)</td>
<td></td>
</tr>
</tbody>
</table>

© 2011 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 270*.

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 286*. 
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.

<table>
<thead>
<tr>
<th>Definition Rule</th>
<th>Name</th>
<th>Destination Folder</th>
<th>Rename</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Existing</td>
<td>Revenue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use Existing</td>
<td>Total</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use Existing</td>
<td>Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use Existing</td>
<td>Average</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use Existing</td>
<td>Minimum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. To change the conflict resolution action for an object, double-click in the Destination 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.
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.

To remove an object from the update package, select the object and click Remove.

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 294. For information about the ACL conflict resolution options, see Conflict resolution and access control lists, page 273.

To create a text file containing a list of the objects in the update package and their conflict resolution actions, click Export.

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.

**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 255.

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 284.*

• **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 12, Automating Administrative Tasks with Command Manager.*

---

**To import an update package from Object Manager**

Before importing any project security update packages, you must import the associated configuration update package.

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 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 295.*

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\Common Files\MicroStrategy`. Use the following parameters:

Only Standard Authentication and Windows Authentication are supported by the Import Package utility.

<table>
<thead>
<tr>
<th>Effect</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import package into this project source (required)</td>
<td><code>n ProjectSourceName</code></td>
</tr>
</tbody>
</table>
| 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)       | `f PackageLocation` |

**Note:** The location must be specified relative to the Intelligence Server machine, not relative to the machine running the Import Package utility.
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 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

When you are ready to import the update package, call the Project Merge executable, `projectmerge.exe`, with the following parameters:

<table>
<thead>
<tr>
<th>Effect</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use this XML file to import an update package (required)</td>
<td><code>-f Filename.xml</code></td>
</tr>
<tr>
<td>Log into the project source with this password (the login ID to be used is stored in the XML file)</td>
<td><code>-sp Password</code></td>
</tr>
<tr>
<td>Log into the project with this password (the login ID to be used is stored in the XML file)</td>
<td><code>-smp Password</code></td>
</tr>
<tr>
<td>Suppress status updates (useful for importing an update package in the background, so that the status window does not appear)</td>
<td><code>-sup</code></td>
</tr>
</tbody>
</table>

To import an update package using Command Manager

Call a Command Manager script that contains the following command:

```plaintext
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

```plaintext
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.)

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 Desktop, 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 31*.

For more detailed information about updating the project schema, see the *Optimizing and Maintaining your Project* chapter in the *MicroStrategy 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 315.*

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 254.*

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 302.*

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 246.*
• 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 MicroStrategy 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 278.

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 297
• Merging projects with the Project Merge Wizard, page 298
• Running Project Merge from the command line, page 300
• Scheduling a project merge, page 302
• Resolving conflicts when merging projects, page 303

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
Merging projects to synchronize objects

Managing Your Projects System Administration Guide

Merging projects to synchronize objects © 2011 MicroStrategy, Inc.

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 304.

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 303.

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 300) or to schedule a merge (see Scheduling a project merge, page 302).

Before you use Project Merge in a server (three-tier) environment, check the project source time out setting. In Desktop, 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 31.*

1. From the Windows **Start** menu, point to **Programs**, then **MicroStrategy**, then **Object Manager**, 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 303.*

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 the 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 the 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.
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 command prompt. For information on running multiple sessions, see *Multiple project merges from the same project, page 301*.

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description and use</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f[ ]</code></td>
<td>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: <code>-f c:\files\merge.xml</code></td>
</tr>
<tr>
<td><code>-sp[ ]</code></td>
<td>Password for SOURCE Project Source. (The login ID to be used is stored in the XML file.) Example: <code>-sphello</code></td>
</tr>
<tr>
<td><code>-dp[ ]</code></td>
<td>Password for DESTINATION Project Source. (The login ID to be used is stored in the XML file.) Example: <code>-dphello</code></td>
</tr>
<tr>
<td><code>-smp[ ]</code></td>
<td>Password for SOURCE metadata. (The login ID to be used is stored in the XML file.) Example: <code>-smphello</code></td>
</tr>
<tr>
<td><code>-dmp[ ]</code></td>
<td>Password for DESTINATION metadata. (The login ID to be used is stored in the XML file.) Example: <code>-dmphello</code></td>
</tr>
<tr>
<td><code>-sup</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>-MD</code></td>
<td>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.</td>
</tr>
</tbody>
</table>
### A sample command using this syntax is provided below. The command assumes that “hello” is the password for all of 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.

```bash
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:

```bash
projectmerge -f "c:program files\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 255*. For detailed instructions on how to manually lock and unlock projects, see the Desktop Help (press F1 from within Desktop).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description and use</th>
</tr>
</thead>
<tbody>
<tr>
<td>-SU</td>
<td>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). <strong>Note:</strong> Do not use this switch if the Project Merge configuration XML contains an instruction to update the schema.</td>
</tr>
<tr>
<td>-lto</td>
<td>Take ownership of any metadata locks that exist on the source or destination projects. For more information about metadata locking, see <em>Locking projects, page 255</em>.</td>
</tr>
<tr>
<td>-h</td>
<td>Displays help and explanations for all of the above parameters.</td>
</tr>
</tbody>
</table>
To do this, you must modify the Project Merge XML file, and then make a copy of it for each session you wish 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:
   ```xml
   <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 Desktop Help (press F1 from within Desktop).
6. Manually lock the destination projects at the configuration level. For detailed steps on locking projects manually, see the Desktop Help (press F1 from within Desktop).
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 300*. 
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 300.

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 page 258.) 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 objects, which, in turn, override both object type rules and object category 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.
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.

<table>
<thead>
<tr>
<th>Action</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing</td>
<td>No change is made to the destination object. The source object is not copied.</td>
</tr>
<tr>
<td>Replace</td>
<td>The destination object is replaced with the source object.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: 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 child objects are added to it, thus changing the version ID and modification times during the copy process.</td>
</tr>
<tr>
<td>Keep both</td>
<td>No change is made to the destination object. The source object is duplicated in the destination location.</td>
</tr>
<tr>
<td>Use newer</td>
<td>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.</td>
</tr>
<tr>
<td>Use older</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 256.

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.
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 256.

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 246.

• 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 256.

1 From the Windows Start menu, point to Programs, then MicroStrategy, then Object Manager, and then select Object Manager. Object Manager opens.

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 256.
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 240.

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 Desktop 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 Desktop interface.)

To search for objects and save the results in a text file

1 In MicroStrategy Desktop, 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 Desktop, consult the Help. (Click Help or press F1.)

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\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 256.

1 From the Windows Start menu, point to Programs, then MicroStrategy, then Object Manager, 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.
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 MicroStrategy 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 MicroStrategy 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 MicroStrategy 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 309
- Deleting all unused managed objects, page 310

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.

To delete managed objects one-by-one

1. In Desktop, 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 Desktop, 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 *MicroStrategy 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 Desktop, from the Administration menu, select Projects, and then select Delete unused managed objects.
**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 313*
- *Monitoring system activity: Change journaling, page 315*
- *Monitoring system usage: Intelligence Server statistics, page 323*
- *Additional monitoring tools, page 338*

**MicroStrategy system monitors**

You can monitor various aspects of your MicroStrategy system from within Desktop. 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.

<table>
<thead>
<tr>
<th>For information about monitoring...</th>
<th>See...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects loaded on Intelligence Server, or on all nodes of the cluster</td>
<td>Managing and monitoring projects, page 28</td>
</tr>
<tr>
<td>Projects loaded on specific nodes of the cluster</td>
<td>Managing your projects across nodes of a cluster, page 446</td>
</tr>
<tr>
<td>Jobs that are currently executing</td>
<td>Monitoring currently executing jobs, page 55</td>
</tr>
<tr>
<td>Users that are currently connected to Intelligence Server</td>
<td>Monitoring users’ connections to projects, page 68</td>
</tr>
<tr>
<td>Active and cached database connections</td>
<td>Monitoring database instance connections, page 11</td>
</tr>
<tr>
<td>Report and document caches</td>
<td>Monitoring result caches, page 466</td>
</tr>
<tr>
<td>History List messages</td>
<td>Monitoring History List messages, page 503</td>
</tr>
<tr>
<td>Intelligent Cubes, whether or not they are loaded on Intelligence Server</td>
<td>Managing Intelligent Cubes: Intelligent Cube Monitor, page 524</td>
</tr>
</tbody>
</table>

**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 79*.

- 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 69*.

**To view a system monitor**

1. In Desktop, 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.
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.

### 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 Desktop or 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.

To enable or disable change journaling for a project source

1. In Desktop, 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.


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 Desktop, 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.

**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 Desktop Preferences dialog box.

To disable the requests for change journaling comments

1 Open MicroStrategy Desktop.

2 From the **Tools** menu, select **Desktop Preferences**. The Desktop 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 Desktop 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 Desktop, expand Administration, then expand System Monitors, and then select Change Journal Transactions. The change journal entries are listed in the main window of Desktop.

You must have the Audit Change Journal privilege to view the change journal.

To view the detailed information for a given change journal entry, double-click that entry. Each entry contains the following information:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object name</td>
<td>The name of the object that is changed.</td>
</tr>
<tr>
<td>Object type</td>
<td>The type of object changed. For example, Metric, User, or Server Definition.</td>
</tr>
<tr>
<td>User name</td>
<td>The name of the MicroStrategy user that made the change.</td>
</tr>
<tr>
<td>Transaction timestamp</td>
<td>The date and time of the change, based on the time on the Intelligence Server machine.</td>
</tr>
<tr>
<td>Transaction type</td>
<td>The type of change and the target of the change. For example, Delete Objects, Save Objects, or Enable Logging.</td>
</tr>
<tr>
<td>Transaction source</td>
<td>The application that made the change. For example, Desktop, Command Manager, MicroStrategy Web, or Scheduler.</td>
</tr>
<tr>
<td>Project name</td>
<td>The name of the project that contains the object that was changed.</td>
</tr>
<tr>
<td>Note: If the object is a configuration object, the project name is listed as &lt;Configuration&gt;</td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>Any comments entered in the Comments dialog box at the time of the change.</td>
</tr>
<tr>
<td>Object ID</td>
<td>The object’s GUID, a unique MicroStrategy system identifier.</td>
</tr>
<tr>
<td>Machine name</td>
<td>The name of the machine that the object was changed on.</td>
</tr>
<tr>
<td>Change type</td>
<td>The type of change that was made. For example, Create, Change, or Delete.</td>
</tr>
<tr>
<td>Transaction ID</td>
<td>A unique 32-digit hexadecimal number that identifies this change.</td>
</tr>
<tr>
<td>Session ID</td>
<td>A unique 32-digit hexadecimal number that identifies the user session in which the change was made.</td>
</tr>
<tr>
<td>Link ID</td>
<td>For internal use only.</td>
</tr>
</tbody>
</table>

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 only displays and exports the last 1000 entries. You can increase this number in the **Browsing** category of the Desktop Preferences dialog box. Viewing more entries may make the browsing and exporting process take longer.

<table>
<thead>
<tr>
<th>To increase the number of displayed or exported change journal entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 In Desktop, from the <strong>Tools</strong> menu select <strong>Desktop Preferences</strong>. The Desktop Preferences dialog box opens.</td>
</tr>
<tr>
<td>2 In the <strong>General</strong> category, select <strong>Browsing</strong>.</td>
</tr>
<tr>
<td>3 In the <strong>Maximum number of monitoring objects displayed per page</strong> field, specify the maximum number of change journal entries to display at once.</td>
</tr>
<tr>
<td>4 In the <strong>Maximum number of transactions retrieved per metadata change journaling search</strong> field, specify the maximum number of change journal entries to export.</td>
</tr>
<tr>
<td>5 Click <strong>OK</strong>. The dialog box closes and your changes are saved.</td>
</tr>
</tbody>
</table>

### Searching the change journal for relevant entries

Since the change journal records every transaction, finding the relevant records can sometimes be daunting. To make searching the change journal easier, you can filter it so that you only see relevant entries.
For example:

- To find out when certain users were given certain permissions, you can view only entries related to Users.
- To discover which user made a change that caused a report to stop executing correctly, you can view only the entries related to that report.

You can also quickly filter the entries so that you see only the entries for a specific object, or only 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**


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 only 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 only, 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 only the changes to this object, select Filter view by object.
   - To see only 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\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 Desktop, 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.
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 321.

To purge the change journal for all projects in a project source

1. In Desktop, 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 prior to this date and time will be deleted from the change journal.
4. To purge data for all projects, select the Apply to all projects check box. To purge only 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 prior to 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 Desktop, 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 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 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.

5 Click **OK**. The Project Configuration Editor closes.

---

**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 323*
- *Best practices for recording Intelligence Server statistics, page 328*
- *Configuring Intelligence Server to log statistics, page 330*
- *Improving database performance by purging statistics data, page 336*

MicroStrategy Enterprise Manager can assist you in analyzing 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 Chapter 14, *Analyzing System Usage with Enterprise Manager*.

---

**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 326*).

The statistics that are logged for each project are set in the Project Configuration Editor, in the **Statistics** category, in the **General** subcategory. The options are as follows:

<table>
<thead>
<tr>
<th>Statistics logging option</th>
<th>Statistics logged</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic statistics</td>
<td>User session and project session analysis. This option must be selected for any statistics to be logged.</td>
</tr>
<tr>
<td>Report job steps</td>
<td>Detailed statistics on the processing of each report.</td>
</tr>
<tr>
<td>Document job steps</td>
<td>Detailed statistics on the processing of each document.</td>
</tr>
<tr>
<td>Report job SQL</td>
<td>The generated SQL for all report jobs. <strong>Warning</strong>: This option can create a very large statistics table. Only select this option when you specifically need the job SQL data.</td>
</tr>
<tr>
<td>Report job tables/columns accessed</td>
<td>Data warehouse tables and columns accessed by each report.</td>
</tr>
<tr>
<td>Prompt answers</td>
<td>Detailed statistics on any prompts that are answered.</td>
</tr>
<tr>
<td>Subscription deliveries</td>
<td>Detailed statistics on reports and documents that are delivered via subscriptions. For more information about subscriptions, see <em>Scheduling reports and documents: Subscriptions, page 553</em>.</td>
</tr>
<tr>
<td>Inbox messages</td>
<td>Detailed statistics on reports and documents that are delivered to the History List (Inbox). For more information about the History List, see <em>Saving report results: History List, page 488</em>.</td>
</tr>
</tbody>
</table>

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 a large number of 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 744*.

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 IS_PERF_MON_STATS table in the statistics database.

---

**To configure the performance counters to record information in the statistics database**

1. Open the Diagnostics and Performance Logging Tool.
   - From Desktop: 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 **Desktop Preferences**. In the **General** category, in the **Advanced** subcategory, select the **Show Diagnostics Menu Option** check box and click **OK**.
   
   - In Windows: From the **Start** menu, point to **Programs**, then **MicroStrategy**, then **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. Make sure the **Use Machine Default Configuration** check box is cleared so that your logging settings are not overridden by the default settings.

4. Select the **Performance Configuration** tab.
5 In the **Statistics** column, select the check boxes for the counters that you want to log to the statistics database.

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 a memory buffer. Every 10 seconds, the contents of the buffer are copied to a second buffer, and then written from this second buffer into the statistics database. Using two buffers avoids simultaneous read and write cycles in the memory buffer, which could otherwise occur during periods of heavy logging activity. For a detailed examination of the tables in the statistics database, see the **Statistics Data Dictionary** in the *Supplemental Reference for System Administration*.

![Warning icon]

If you are using Enterprise Manager to monitor your statistics, the database that hosts the statistics tables also contains the Enterprise Manager data warehouse. For more information about the structure of the Enterprise Manager data warehouse, see the **Enterprise Manager Data Dictionary** in the *Supplemental Reference for System Administration*.

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 the purpose of logging statistics. However, the maximum number of database connections is typically only seen in high concurrency environments.

![Warning icon]

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 projects loaded on each node has 20 connections to the warehouse (10 for each node). Even if the same ten projects are loaded on both nodes, there are 20 database connections.
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 given project source must be configured to log statistics individually. This configuration is called Complete Session Logging. It allows some projects to log statistics to one database and some projects to log to another database.

The Enterprise Manager data warehouse must be in the same database as the statistics database for a given 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 databases. A separate Enterprise Manager project must be configured for each statistics database.

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 323.

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.

---

To log all statistics from a project source to the same database

1. In Desktop, right-click the project source and select Configure MicroStrategy Intelligence Server. The Intelligence Server Configuration Editor opens.
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 will be logged to the database instance specified for this project.
5. Click OK. The Intelligence Server Configuration Editor closes.

---

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 database. 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 327.

- Use the sizing guidelines (see Sizing guidelines for the statistics database, page 329) to plan how much hard disk space you need for the statistics database.

- Back up your statistics database regularly, and retain only as much data as is required for your analysis. You can purge the statistics database on a regular basis to decrease table size and improve performance. For instructions on how to purge statistics, see Improving database performance by purging statistics data, page 336.
• Use Enterprise Manager to monitor and analyze the statistics information. For more information about Enterprise Manager, see Chapter 14, Analyzing System Usage with Enterprise Manager.

Sizing guidelines for the statistics database

The following guidelines can assist you in determining how much space you need for the statistics database. These guidelines are only for planning purposes; MicroStrategy recommends that you monitor the size of your statistics database and adjust your hardware requirements accordingly.

• When the Basic Statistics, Report Job Steps, Document Job Steps, Report SQL, Report Job Tables/Columns Accesssed, and Prompt Answers statistics are logged, a single user executing a single report increases the size of the statistics database 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% of the reports that are executed return fewer than 1,000 cells, the average report increases the size of the statistics database by less than 10 kilobytes.

• When the Subscription Deliveries and Inbox Messages statistics are logged, each subscription that is delivered increases the size of the statistics database 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 size of the database by an average of 0.4 kilobytes. You can control the growth of this table by specifying what counters to log and how often to log each counter. For more information on logging performance counters to the statistics database, including instructions, see Recording performance counters in the statistics tables, page 324.

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%, 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

Based on 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 330).

2. Create the statistics tables in the statistics database (see *Creating statistics tables in the statistics database*, page 331).

3. Configure your project to log statistics to the specified database (see *Setting the statistics database instance for a project*, page 332).

4. Choose what statistics to log from that project (see *Specifying which statistics to log*, page 335).

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 the Data Source Name (DSN) for it. 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 *Chapter 14, Analyzing System Usage with Enterprise Manager*.

- 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 327*, or see the readme file for Intelligence Server.
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 Programs, then MicroStrategy, then Tools, then select Connectivity Wizard. For detailed instructions on using the Connectivity Wizard, see the MicroStrategy 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


   • Windows: From the Windows Start menu, point to Programs, then MicroStrategy, 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.


3. Select the Statistics Tables option and clear all other options. Click Next. The Statistics Tables page opens.
4 From the DSN drop-down list, select the DSN for the statistics database.

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 login that you specify must have permission to create and drop tables in the database, and permission to create views.

6 Click Advanced.

7 In the Script field, the default script is displayed. The selected script depends on the database type that you specified previously. To select a different script, click ... (the Browse button) to browse to and select a script that corresponds to the DBMS that you are using. The default location for these scripts is:

\Program Files\Common Files\MicroStrategy\n
<table>
<thead>
<tr>
<th>Data Warehouse</th>
<th>SQL Script to Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server</td>
<td>statistics_SQLServer.sql</td>
</tr>
<tr>
<td>Oracle</td>
<td>statistics_Oracle.sql</td>
</tr>
<tr>
<td>DB2</td>
<td>statistics_Db2.sql</td>
</tr>
<tr>
<td>Teradata</td>
<td>statistics_Teradata.sql</td>
</tr>
<tr>
<td>Sybase</td>
<td>statistics_Sybase.sql</td>
</tr>
</tbody>
</table>

8 Click Next.

9 The final screen shows a summary of your choices. To create the statistics tables, click Finish.

**Setting the statistics database instance for a project**

Once the statistics database 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 more information about single instance session logging, see *Logging all statistics from a project source to the same database, page 327.*

**To set up a project to log statistics**

1. In Desktop, log in to the server (three-tier) project source containing the projects that you want to log statistics for. 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 database. Click New. The Database Instances dialog box opens.
5 In the **Database instance name** field, type in a name for the statistics logging database instance.

6 From the **Database connection type** drop-down list, select the database type and version that corresponds to the statistics database DBMS.

   If you are using Oracle 8i R3, select “Oracle 8i R2/R3.”

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.

9 From the **ODBC Data Sources** list, select the Data Source Name used to connect to the statistics database.

10 If your database supports parameterized queries, you can improve the efficiency of the statistics logging process by enabling parameterized queries in the statistics database instance. 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. Each time you click **OK**, make sure your new database login and database connection are selected before clicking **OK**.

15 In the **Database Instances** category, select the **Statistics** subcategory.

16 From the **Statistics DB Instance** drop-down list, select your new statistics database instance.

17 Click **OK**. The Project Configuration Editor closes and your selections are saved.
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 323.

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 327) 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 detailed information on how to log performance information, including instructions, see Recording performance counters in the statistics tables, page 324.

To specify which statistics to log

1. In Desktop, log in to the project source containing the project that you want to log statistics for. 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.

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 Desktop, 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.

---

**Improving database performance by purging statistics data**

The statistics database collects a great deal of data. Over time some of this data can become outdated and no longer relevant to your analysis. When this happens, MicroStrategy recommends that you purge the outdated statistics to ensure optimum performance.

Statistics can be purged from the statistics table at the project level or at the server (project source) level. In either case, you must specify a date range for the purge.

All database purge processes are run as a database transaction, and the transaction is rolled back in the event of any failure, to ensure that the statistics database is in a consistent state. To keep the size of these transactions small, it is recommended that you set many small time windows for purge operations rather than specifying a single large time window, depending on the size of your statistics database. For example, for a very large database, the time window can be reasonably set to one month; for a small database a more appropriate time span could be one year.

If the delete operation against a particular statistics table cannot be performed within the specified purge timeout duration, a SQLCancel command is issued by Intelligence Server. When this happens, the statistics data is not purged and an error message is displayed. In this case you may need to increase the purge timeout or decrease the purge window.

---

**To purge statistics for a particular project**

The IS_SESSION_STATS table does not contain any project-level information and is therefore not affected by purges at the project level.

1 In Desktop, right-click the project and select **Project Configuration**. The Project Configuration Editor opens.

2 Select the **Statistics** category, and the **Purge** subcategory beneath it.
3 In the **Select dates** drop-down lists, select the date range for which to purge statistics.

4 In the **Purge timeout (sec.)** field, specify the timeout setting in seconds. The server uses this during the purge operation. The server issues a single SQL statement to purge each statistics table, and the time out setting applies to each individual SQL statement issued during the purge operation.

5 Click the **Advanced** button. A list of statistics is displayed.

6 Select the statistics that you want to purge. For an explanation of these statistics, see *Overview of Intelligence Server statistics, page 323*.

7 If you are using clustered Intelligence Servers:
   - To only purge project statistics logged by the current Intelligence Server, select the **Only purge statistics logged from the current Intelligence Server machine** check box. Statistics logged by other Intelligence Servers are ignored.
   - To purge project statistics logged by any of the clustered Intelligence Servers, clear the **Only purge statistics logged from the current Intelligence Server machine** check box.

8 Click **Purge Now**. The statistics for the specified dates are deleted from the statistics database.

Statistics at the project level can also be purged with a Command Manager script, using the **PURGE STATISTICS** command in the Project Configuration area. For more information about using Command Manager, including sample scripts, see *Chapter 12, Automating Administrative Tasks with Command Manager*.

---

**To purge statistics for all projects on a particular project source at once**

1 In Desktop, right-click the project source and select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration dialog box opens.

2 Select the **Statistics** category, and the **Purge** subcategory beneath it.

3 In the **Select dates** drop-down lists, select the date range for which to purge statistics.
4 In the **Purge timeout (sec.)** field, specify the time out setting in seconds. The server uses this during the purge operation. The server issues a single SQL statement to purge each statistics table, and the time out setting applies to each individual SQL statement issued during the purge operation.

5 Click the **Advanced** button. A list of statistics is displayed.

6 Select the statistics that you want to purge. For an explanation of these statistics, see *Overview of Intelligence Server statistics, page 323*.

7 If you are using clustered Intelligence Servers:
   - To only purge statistics logged by the current Intelligence Server, select the **Only purge statistics logged from the current Intelligence Server machine** check box. Statistics logged by other Intelligence Servers are ignored.
   - To purge statistics logged by any of the clustered Intelligence Servers, clear the **Only purge statistics logged from the current Intelligence Server machine** check box.

8 Click **Purge Now**. The statistics for the specified dates are deleted from the statistics database.

### 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 740*. 
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 13, Verifying Reports and Documents with Integrity Manager.

Enterprise Manager

MicroStrategy Enterprise Manager helps you analyze Intelligence Server statistics. Enterprise Manager provides a pre-built MicroStrategy project with over a hundred reports and dashboards covering all aspects of Intelligence Server operation. You can also use Enterprise Manager’s pre-built facts and attributes to create your own reports, so that you can have immediate access to the performance and system usage information that you need.

For more information about Enterprise Manager, see Chapter 14, Analyzing System Usage with Enterprise Manager.

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 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 15, Maintaining your MicroStrategy System with Health Center.
Windows Performance Monitor

The Windows Performance Monitor is not part of MicroStrategy, but 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 353.*
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 chapter include:

- Tuning overview and best practices, page 342
- Designing system architecture, page 347
- Managing system resources, page 352
- Managing user sessions, page 371
- Governing requests, page 379
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 concurrent 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 based on 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 343
- Configuring the system design, page 343
- Best practices for tuning your system, page 346
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.

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 actively logged in.
- 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.

UNIX and Linux systems allow processes and applications to run in a virtual environment. MicroStrategy 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 371*) and executing jobs (see *Managing job execution, page 386*).
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 74.*

---

**To access the Intelligence Server Configuration Editor**

1. In Desktop, 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 Desktop, 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 MicroStrategy Project Design Guide.
- When configuring your network, follow the MicroStrategy best practices as outlined in Network configuration best practices, page 350.
- 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 353.
- Use Intelligence Server’s Memory Contract Manager to manage memory usage, as described in Governing Intelligence Server memory use with Memory Contract Manager, page 360.
- Use MicroStrategy system privileges to restrict users’ access to certain features, as described in Governing user profiles, page 377.
- 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 390.
- Enable Intelligence Server thread balancing, as described in Intelligence Server thread balancing, page 394.
- 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 404.
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 347*)

• The physical location of machines relative to each other and the amount of bandwidth between them (see *How the network can affect performance, page 348*)

• Whether you cluster several Intelligence Servers together and what benefits you can get from clustering (see *How clustering can affect performance, page 351*)

**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.

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 MicroStrategy 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Protocol</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HTTP</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>TPC/IP</td>
<td>XML requests are sent to Intelligence Server. XML report results are incrementally fetched from Intelligence Server.</td>
</tr>
<tr>
<td>3</td>
<td>TCP/IP</td>
<td>Requests are sent to Intelligence Server. (No incremental fetch is used.)</td>
</tr>
<tr>
<td>4</td>
<td>TCP/IP</td>
<td>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.</td>
</tr>
<tr>
<td>5</td>
<td>TCP/IP</td>
<td>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 427).</td>
</tr>
<tr>
<td>6</td>
<td>ODBC</td>
<td>Object requests and transactions to metadata. Request results are stored locally in Intelligence Server object cache.</td>
</tr>
<tr>
<td>7</td>
<td>ODBC</td>
<td>Complete result set is retrieved from database and stored in Intelligence Server memory and/or caches.</td>
</tr>
</tbody>
</table>

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 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.

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% (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% for your system.

To calculate the network capacity utilization percent, take the total capacity, in terms of bits/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
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 353)
- Physical disk characteristics (Physical disk, page 354)
- The amount of memory (Memory, page 355)

The MicroStrategy Installation and Configuration Guide contains detailed information about small, medium, and large configurations.
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, and export that data in various formats including Microsoft Excel and SQL. You can also use it to access specific MicroStrategy performance counters.

For more information on Windows Performance monitor, see http://www.microsoft.com/.

To access the Windows Performance Monitor

From the Windows Start menu, point to Programs, then MicroStrategy, then Intelligence Server, and then choose Microsoft Performance Monitor.

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%, 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 353.

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 together, 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 224.

Physical disk

If the physical disk is utilized 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% 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 355.

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 given 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 353.*

**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 and user address space, page 355*
- *Monitoring memory use with Performance Monitor, page 357*
- *How much memory does Intelligence Server use when it starts up?, page 358*
- *How does Intelligence Server use memory after it is running?, page 359*
- *Governing Intelligence Server memory use with Memory Contract Manager, page 360*

**Memory limitations: virtual memory and user address space**

The memory used by Intelligence Server is limited by two factors:

- The machine’s virtual memory (see *Virtual Memory, page 356*)
- The user address space used by the Intelligence Server process (see *User address space, page 356*)
Virtual Memory

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.

Private bytes are the bytes of virtual memory that are allocated to a given 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.

User address space

User address space (UAS) is Intelligence Server’s second memory limitation. It is independent of virtual memory, and is of finite size. By definition, in a 32 bit operating system, the size of each process’s address space (such as the Mstrsvr.exe Intelligence Server application) is defined by the number of bytes that may be addressed in 32 bits, which is $2^{32}$, or 4GB. The Windows operating system divides this virtual address space into two parts: System Address Space (SAS), which is reserved for the operating system’s use; and UAS. The UAS is, in this case, for Intelligence Server to store data and code.

Windows 2003 allocates 2 GB each to the UAS and SAS. However, when using 4GT mode in Windows 2003 Advanced Server (or higher edition), the operating system allocates 3 GB to UAS and 1 GB to SAS. For instructions on how to enable 4GT, see Microsoft’s support website, or see MicroStrategy Tech Note TN6875.
The virtual address space used by a process does not represent the actual virtual memory. Instead, the system maintains a page map for each process. A page map is an internal data structure used to translate virtual addresses into corresponding physical (RAM and page file) addresses. For this reason, the total virtual address space of all processes is much larger than the total virtual memory available.

The limit associated with Intelligence Server 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). This value is called the process’s \textit{virtual bytes}. Memory depletion is usually caused by running out of virtual bytes.

\textbf{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 \textit{Managing system memory and resources: Windows Performance Monitor, page 353}.

The two memory-related counters you should log with Performance Monitor are \textbf{Private Bytes} and \textbf{Virtual Bytes} for the Intelligence Server process (\texttt{Mstrsvr.exe}). A sample log of these two counters (along with others) for Intelligence Server is shown in the diagram below.
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 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 353.

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 9, 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 10, Managing Intelligent Cubes.
• 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 371* and *Saving report results: History List, page 488*.

• 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 379* and *Results processing, page 394*.

• 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 8, 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 11, 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 will cause the system to approach a state of memory depletion. When enabled, MCM grants or denies requests for memory from tasks within Intelligence Server. The requests are granted or denied based on user-configured limits on the amount of memory Intelligence Server is allowed to use. Since MCM is a component within Intelligence Server, it does not manage the actual memory used by Intelligence Server itself.

The types of requests governed by MCM are:

• 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.

**Using the Memory Contract Manager**

The MCM settings can be found in the Intelligence Server Configuration Editor, in the **Governing: Memory Settings** category.

![Memory Contract Manager Image]

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, while a later 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 current memory usage to exceed the **Maximum use of virtual address space** or **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 information on how the low water mark is calculated, see *Memory water marks, page 365*. For information about how MCM handles memory request idle mode, see *Memory request idle mode, page 366*.

The **Maximum use of virtual address space** setting is only used in 32-bit operating systems. In a 64-bit operating system, the amount of memory that can be reserved with this setting is so large as to be meaningless. Use the **Minimum reserved memory** setting to control the amount of memory available for Intelligence Server.

The **Memory request idle time** is the longest amount of 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 already been contracted out to the tasks.

A memory request is only 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 current Intelligence Server memory usage and the **Maximum use of virtual address space** and **Minimum reserved memory** settings. For a more detailed explanation of the memory water marks, see *Memory water marks, page 365*.
- It is smaller than 80% 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.

Key:
- I-Server = Intelligence Server
- PB = Private Bytes
- VB = Virtual Bytes
- HWM = High Watermark
- LWM = Low Watermark
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 355.

- 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. Since 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 $[\text{HWM} - (1.05 \times (\text{Intelligence Server Private Bytes} + \text{Outstanding Contracts})]$.

- In memory request idle mode, the maximum request size is based on the low water mark. The formula is $[\text{LWM} - (1.05 \times (\text{Intelligence Server Private Bytes} + \text{Outstanding Contracts})]$.

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.

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 366*.

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% 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, then 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, since Intelligence Server is completing its existing contracts and releasing memory, it is possible that the next contract request submitted will be below the LWM. In this case, MCM accepts the request and resumes normal operation, without having to restart Intelligence Server.

When MCM forces Intelligence Server to restart due to 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\MicroStrategy\Intelligence Server\.

MicroStrategy recommends setting the Memory request idle time to slightly longer than the amount of time that 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 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 Chapter 14, Analyzing System Usage with Enterprise Manager.
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, all requests that would push the total memory used above the low water mark are denied.

In the example above, request C falls above the low water mark. Since Intelligence Server is in memory request idle mode, this request will be denied unless Intelligence Server releases memory from elsewhere, such as other completed contracts.

Request D is below the low water mark, so it will be granted. Once it has been granted, Intelligence Server switches out of request idle mode and resumes normal operation.

If Intelligence Server does not receive any requests for an amount of memory below 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 MicroStrategy Intelligence Server virtual address space.

![Diagram of Virtual Byte Depletion]

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% 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. See below for more details about each setting.

- **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 (Web, Desktop, 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 has the potential to consume additional resources depending on the actions that the user takes while she is connected.

The number of concurrent users in a system (those actually executing reports and using the system) is considered a different category of user from active users (those simply logged in).

This section covers:

- How the active users and user sessions on your system use system resources just by logging in to the system (see Governing active users, page 372)
- 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 374)
- 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 377)

You can track the users who are currently connected to the system with the User Connection Monitor. For details about how to use this system monitor, see Monitoring users’ connections to projects, page 68.
**Governing active 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 potentially put on the system because each session can run multiple jobs.

To help control the potential load that user sessions can put on the system, you can limit the number of user sessions allowed for each project and for Intelligence Server. Also, both Desktop 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 of 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 specific 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, Desktop, 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: 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: 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 a single 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: 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 MicroStrategy Desktop (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.

To specify this setting for Desktop, in the Intelligence Server Configuration Editor, select the **Governing: 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: 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 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.*

**Governing user resources**

User sessions consume system resources when users log in to the system, particularly when they use the History List and, in MicroStrategy Web, the Working Set. This section discusses these two features and how you can govern them.

**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 has the potential to consume a great deal 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 about scheduling this task, including instructions, see *Scheduling administrative tasks, page 560.*
• 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 488.*

**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 that 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 within 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: 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.

- If you set this limit too low, and you do not have enough hard disk space to handle the amount of disk swapping, then during peak usage periods reports may fail to execute because they cannot write to memory or to disk.

If a user session has an open job, the user session does not close, and that job’s report instance is not removed from the Working set, until 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 383.
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 within the system, they can potentially affect the performance of the system. 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 79.

Subscription-related privileges

Allowing users to subscribe to reports to be run at a later date can potentially affect system performance. You can limit the use of subscriptions by using the Web Scheduled Reports and Schedule Request privileges.

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 predetermined 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 553. For information about Distribution Services, see Scheduling deliveries to email, file, and printer using Distribution Services, page 564.

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 374). 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 401*. 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 PDF
• 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 have the potential to 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 407.

**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 has the potential for using 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 380).

Specifically, you can limit:

- The total number of jobs (Limiting the total number of jobs, page 380)
- The number of jobs per project (Limiting the number of jobs per project, page 381)
- The number of jobs per user account and per user session (Limiting the number of jobs per user session and per user account, page 381)
• 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 382)

• The amount of time reports can execute (*Limiting the maximum report execution time*, page 383)

• A report's SQL (per pass) including both its size and the time it executes (*Limiting a report's SQL per pass*, page 385)

• The amount of memory used for Intelligent Cubes (*Governing Intelligent Cube memory usage*, page 532)

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: General** category, and select the **For Intelligence Server 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: 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.

**Limiting the number of jobs per project**

You can limit the number of concurrent jobs that are being processed by Intelligence Server for a given 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: 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: 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 potentially run eight jobs. But if this Jobs per user account limit is set to five, that user can only execute 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.

To specify this setting, in the Project Configuration Editor for the project, select the Governing: 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 390*).

To specify this limit setting, in the Project Configuration Editor for the project, select the **Governing: 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, use the Project Configuration Editor, select the **Governing: 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.
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 38.
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 Intelligence Server Configuration Editor, open the Governing: 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. (See the Desktop Help for details.)

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. (See the Desktop Help for details.)
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.

This section discusses the different ways you have of managing job execution. These include:

- Managing database connection threads, page 386
- Prioritizing jobs, page 390
- Results processing, page 394 (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.

![Optimal use of resources](image)

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 11. 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 one 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 only use 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 390.

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.

**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 385). 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 cancelled 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 390), and you can limit the amount of time an inactive database connection remains open (see *Connection idle timeout*, page 390).

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.

- 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 on a first-come, first-served basis. 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, while 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 387.

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, based on one or more of the following variables:

- **Request type**: Report requests and element requests can have different priority (Prioritizing jobs by request type, page 392).

- **Application type**: Jobs submitted from different MicroStrategy applications, such as Desktop, Scheduler, MicroStrategy Web, or Narrowcast Server, are processed according to the priority that you specify (Prioritizing jobs by MicroStrategy application type, page 392).

- **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 393).

- **Cost**: Jobs with a higher resource cost are processed according to the priority that you specify (Prioritizing jobs by report cost, page 393). 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 394).

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 Desktop, 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.

**Prioritizing jobs by MicroStrategy application type**

You can assign a different priority to jobs submitted from MicroStrategy Desktop, 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 MicroStrategy Desktop 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 particular 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 394.

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 Desktop, 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.

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 394*)
- The size of the report (see *Limiting the maximum report size, page 395*)
- Whether the report is an Intelligent Cube (see *Limiting the size and number of Intelligent Cubes, page 398*)
- Whether the report is imported from an external data source (see *Limiting the memory used during data fetching, page 399*)

**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 MicroStrategy Desktop. The Cache Monitor shows the size of the report results in binary format, which from testing has proven to be 30 – 50% of the actual size of the report instance in memory. For instructions on how to use the Cache Monitor to view the size of a cache, see Monitoring result caches, page 466.

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 395
- Limiting the number of element rows, page 396
- Limiting the number of intermediate rows, page 397

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 within 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: 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 Desktop, 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.
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 Desktop.

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 504.*

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 will incrementally fetch 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 Desktop, in the Project Configuration Editor for that project, expand the **Governing: 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 **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** and **Maximum number of report objects per block** fields.

5. Click **OK**. The new maximum values are saved.

**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 within 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 within 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 could 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: 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 Desktop, 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 can potentially 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.
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 537*.

### 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 potentially 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: 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 386*), 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 400)
- How many XML cells in a result set can be delivered simultaneously (see Limiting the number of XML cells, page 401)

The following settings also govern results delivery:

- The maximum size of a report that can be exported (see Limiting export sizes, page 402 and Limiting the memory consumption for file generation, page 403)
- The number of XML drill paths in a report (see Limiting the total number of XML drill paths, page 404)

## 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 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.

---

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 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 401) 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 400). 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: File Generation** category, then specify the **Maximum number of XML cells**. You must restart Intelligence Server for the new limit to take effect.

### 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 **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 of the formatting data. For more information about exporting reports, see What happens when I export a report from Web?, page 52.

Because Excel export uses significantly more memory than other export formats, you can limit the size of reports exported to Excel from Desktop 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: File Generation category, and specify the Maximum memory consumption for the XML, PDF, Excel, and HTML files.

---

To set the maximum memory consumption for Excel file generation

1. In Desktop, 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 MicroStrategy Advanced Reporting Guide.

To set this limit, in the Intelligence Server Configuration Editor, select the Governing: 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, particularly for large reports. For more information, see XML caches, page 461 and ACLs and personalized drill paths in Web, page 78.

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 405*)
- Subtotals (*Subtotals, page 405*)
- Page-by (*Page-by feature, page 406*)
- Prompt complexity (*Prompt complexity, page 406*)
- Report Services documents (*Report Services documents, page 406*)
- Intelligent Cubes (*Intelligent Cubes, page 407*)

### 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 impact 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 subtotalling 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 Desktop, 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 can potentially 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 impact 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 396*), you should make sure your element caches are being used effectively. For information on managing element caches, including instructions, see *Element caches, page 504*.

### Report Services documents

Report Services documents can contain multiple reports. Executing a single 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 impact of Report Services documents on the system, consider enabling document caching. If the documents are cached on Intelligence Server, less load is placed on the data warehouse and on the Intelligence Server machine. For information about document caching, including instructions, see *Result caches, page 456*. 
Intelligent Cubes

If you have purchased OLAP Services licenses from MicroStrategy, your report designers can create Intelligent Cube reports. These reports allow data to be returned from the data warehouse and stored in Intelligence Server memory, and then shared among multiple reports.

Since Intelligent Cubes must be loaded into memory to be used in reports, they have the potential to 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 534.

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 398.

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 408)
- Project Configuration Editor (Project Configuration Editor, page 412)
- Database connections (Database connection, page 416)
- VLDB settings (VLDB settings, page 417)

**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

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of network threads</td>
<td>Controls the number of network connections available for communication between Intelligence Server and the client, such as Desktop or MicroStrategy Web.</td>
<td>348</td>
</tr>
</tbody>
</table>

**Server definition: Advanced** category in Intelligence Server configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup frequency (min)</td>
<td>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.</td>
<td>485</td>
</tr>
<tr>
<td>Balance MicroStrategy Server threads</td>
<td>Controls whether threads within Intelligence Server are allocated to processes such as object serving, element serving, SQL generation, and so forth that need them most, while less loaded ones can return threads to the available pool.</td>
<td>394</td>
</tr>
<tr>
<td>Cache lookup cleanup frequency (sec)</td>
<td>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.</td>
<td>485</td>
</tr>
<tr>
<td>Project failover latency (min.)</td>
<td>The amount of time (the delay) before the project is loaded on another server to maintain minimum level availability.</td>
<td>449</td>
</tr>
</tbody>
</table>
## Governing: General category in Intelligence Server configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration recovery latency (min.)</td>
<td>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.</td>
<td>450</td>
</tr>
<tr>
<td>Enable performance monitoring</td>
<td>Configures additional MicroStrategy-specific monitors in Windows Performance Monitor.</td>
<td>357</td>
</tr>
<tr>
<td>Maximum number of jobs</td>
<td>The maximum concurrent number of jobs that can exist on an Intelligence Server.</td>
<td>380</td>
</tr>
<tr>
<td>Maximum number of interactive jobs</td>
<td>Limits the number of concurrent interactive (non-scheduled) jobs that can exist on this Intelligence Server. A value of -1 indicates no limit.</td>
<td>380</td>
</tr>
<tr>
<td>Maximum number of scheduled jobs</td>
<td>Limits the number of concurrent scheduled jobs that can exist on this Intelligence Server. A value of -1 indicates no limit.</td>
<td>380</td>
</tr>
<tr>
<td>Maximum number of user sessions</td>
<td>The maximum number of user sessions (connections) for an Intelligence Server. A single user account may establish multiple sessions to an Intelligence Server.</td>
<td>372</td>
</tr>
<tr>
<td>User session idle time (sec)</td>
<td>The time allowed for a Desktop 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.</td>
<td>373</td>
</tr>
<tr>
<td>Web user session idle time (sec)</td>
<td>The time allowed for a Web user to remain idle before his or her session is ended.</td>
<td>373</td>
</tr>
<tr>
<td>For Intelligence Server and History List</td>
<td>Do not include reports submitted as part of a document in the count of jobs for the job limits.</td>
<td>380</td>
</tr>
<tr>
<td>governing, exclude reports embedded in</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Services documents from the counts</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Governing: File Generation category in Intelligence Server configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Generation: Maximum number of XML cells</td>
<td>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.</td>
<td>401</td>
</tr>
<tr>
<td>XML Generation: Maximum number of XML drill paths</td>
<td>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 of the available drill attributes.</td>
<td>404</td>
</tr>
<tr>
<td>XML Generation: Maximum memory consumption for XML (MB)</td>
<td>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.</td>
<td>403</td>
</tr>
<tr>
<td>PDF Generation: Maximum memory consumption for PDF (MB)</td>
<td>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.</td>
<td>403</td>
</tr>
<tr>
<td>Excel Generation: Maximum memory consumption for Excel (MB)</td>
<td>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.</td>
<td>403</td>
</tr>
<tr>
<td>HTML Generation: Maximum memory consumption for HTML (MB)</td>
<td>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.</td>
<td>403</td>
</tr>
</tbody>
</table>

Governing: Memory Settings category in Intelligence Server configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Web request job throttling</td>
<td>A check box that enables the following governors: • Maximum Intelligence Server use of total memory • Minimum machine free physical memory</td>
<td>370</td>
</tr>
<tr>
<td>Maximum Intelligence Server use of total memory (%)</td>
<td>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.</td>
<td>370</td>
</tr>
</tbody>
</table>
### Governor: Working Set category in Intelligence Server configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum RAM for Working Set cache (MB)</td>
<td>The maximum amount of memory that can be used for report instances referenced by messages in the Working Set.</td>
<td>375</td>
</tr>
</tbody>
</table>

### Governor: Import Data category in Intelligence Server configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of connections by priority</td>
<td>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.</td>
<td>386</td>
</tr>
</tbody>
</table>
History Settings: General category in Intelligence Server configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of messages per user</td>
<td>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.</td>
<td>488</td>
</tr>
<tr>
<td>Message lifetime (days)</td>
<td>The length of time before a History List message expires and is automatically deleted. A value of -1 indicates that messages do not expire.</td>
<td>488</td>
</tr>
<tr>
<td>Repository type</td>
<td>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).</td>
<td>488</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of elements to display</td>
<td>The maximum number of attribute elements that can be being retrieved from the data warehouse at one time.</td>
<td>508</td>
</tr>
</tbody>
</table>

Governing: Result sets category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligence Server elapsed time - Interactive reports (sec)</td>
<td>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.</td>
<td>383</td>
</tr>
<tr>
<td>Intelligence Server elapsed time - Scheduled reports (sec)</td>
<td>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.</td>
<td>383</td>
</tr>
<tr>
<td>Final result rows - Intelligent Cubes</td>
<td>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.</td>
<td>395</td>
</tr>
</tbody>
</table>
### 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.

### 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.

### 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.

### All element browsing rows
The maximum number of rows that can be retrieved from the data warehouse for an element request.

### 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.

### 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).

#### Governing: Jobs category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobs per user account</td>
<td>The maximum number of concurrent jobs per user account and project.</td>
<td>381</td>
</tr>
<tr>
<td>Jobs per user session</td>
<td>The maximum number of concurrent jobs a user can have during a session.</td>
<td>381</td>
</tr>
<tr>
<td>Executing jobs per user</td>
<td>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.</td>
<td>382</td>
</tr>
<tr>
<td>Jobs per project - interactive</td>
<td>The maximum number of concurrent ad-hoc jobs that the project can process at a time.</td>
<td>381</td>
</tr>
<tr>
<td>Jobs per project - scheduled</td>
<td>The maximum number of concurrent scheduled jobs that the project can process at a time.</td>
<td>381</td>
</tr>
</tbody>
</table>
### Governing: User sessions category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>User sessions per project</td>
<td>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.</td>
<td>373</td>
</tr>
<tr>
<td>Concurrent interactive project sessions per user</td>
<td>The maximum number of concurrent sessions per user.</td>
<td>373</td>
</tr>
</tbody>
</table>

### Governing: Subscriptions category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum History List subscriptions per user</td>
<td>The maximum number of reports or documents to which a user can be subscribed for delivery to the History List.</td>
<td>559</td>
</tr>
<tr>
<td>Maximum Cache Update subscriptions per user</td>
<td>The maximum number of reports or documents to which a user can be subscribed for updating caches.</td>
<td>559</td>
</tr>
<tr>
<td>Maximum email subscriptions per user</td>
<td>The maximum number of reports or documents to which a user can be subscribed for delivery to an email address (Distribution Services only).</td>
<td>559</td>
</tr>
<tr>
<td>Maximum file subscriptions per user</td>
<td>The maximum number of reports or documents to which a user can be subscribed for delivery to a file location (Distribution Services only).</td>
<td>559</td>
</tr>
<tr>
<td>Maximum print subscriptions per user</td>
<td>The maximum number of reports or documents to which a user can be subscribed for delivery to a printer (Distribution Services only).</td>
<td>559</td>
</tr>
<tr>
<td>Maximum Mobile subscriptions per user</td>
<td>The maximum number of reports or documents to which a user can be subscribed for delivery to a Mobile device (MicroStrategy Mobile only).</td>
<td>559</td>
</tr>
</tbody>
</table>

### Governing: Data Import category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum file size (MB)</td>
<td>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.</td>
<td>539</td>
</tr>
<tr>
<td>Maximum quota per user (MB)</td>
<td>The maximum size of all data import cubes for each individual user.</td>
<td>539</td>
</tr>
</tbody>
</table>
### Caching: Result caches: Storage category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasets - Maximum RAM usage (MBytes)</td>
<td>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.</td>
<td>476</td>
</tr>
<tr>
<td>Datasets - Maximum number of caches</td>
<td>The maximum number of report and dataset caches that the project can have at a time.</td>
<td>470</td>
</tr>
<tr>
<td>Formatted documents - Maximum RAM usage (MBytes)</td>
<td>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.</td>
<td>476</td>
</tr>
<tr>
<td>Formatted documents - Maximum number of caches</td>
<td>The maximum number of document caches that the project can have at a time.</td>
<td>470</td>
</tr>
<tr>
<td>RAM Swap Multiplier</td>
<td>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.</td>
<td>484</td>
</tr>
</tbody>
</table>

### Caching: Result caches: Maintenance category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never expire caches</td>
<td>Determines whether or not caches automatically expire.</td>
<td>485</td>
</tr>
<tr>
<td>Cache duration (Hours)</td>
<td>The amount of time that a result cache remains valid.</td>
<td>485</td>
</tr>
<tr>
<td>Do not override cache expiration settings for reports containing dynamic dates</td>
<td>Select this check box for report caches with dynamic dates to expire in the same way as other report caches.</td>
<td>486</td>
</tr>
</tbody>
</table>

### Caching: Auxiliary caches: Objects category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server - Maximum RAM usage (MBytes)</td>
<td>The amount of memory that Intelligence Server allocates for object caching.</td>
<td>521</td>
</tr>
<tr>
<td>Client - Maximum RAM usage (MBytes)</td>
<td>The amount of memory that Desktop allocates for object caching.</td>
<td>521</td>
</tr>
</tbody>
</table>
Caching: Auxiliary caches: Elements category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server - Maximum RAM usage (MBytes)</td>
<td>The amount of memory that Intelligence Server allocates for element caching.</td>
<td>516</td>
</tr>
<tr>
<td>Client - Maximum RAM usage (MBytes)</td>
<td>The amount of memory that Desktop allocates for object caching.</td>
<td>516</td>
</tr>
</tbody>
</table>

Caching: Subscription Execution category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re-run History List and Mobile subscriptions against the warehouse</td>
<td>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.</td>
<td>562</td>
</tr>
<tr>
<td>Re-run file, email, or print subscriptions against the warehouse</td>
<td>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.</td>
<td>562</td>
</tr>
<tr>
<td>Do not create or update matching caches</td>
<td>Prevents subscriptions from creating or updating caches by default.</td>
<td>562</td>
</tr>
<tr>
<td>Keep document available for manipulation for History List subscriptions only</td>
<td>Retains a document or report for later manipulation that was delivered to the History List.</td>
<td>562</td>
</tr>
</tbody>
</table>

Intelligent Cubes: General category in Project Configuration

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum RAM Usage (MBytes)</td>
<td>The maximum amount of memory used on Intelligence Server by Intelligent Cubes for this project.</td>
<td>537</td>
</tr>
<tr>
<td>Maximum number of cubes</td>
<td>The maximum number of Intelligent Cubes that can be loaded onto Intelligence Server for this project.</td>
<td>537</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of database connection threads</td>
<td>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).</td>
<td>386</td>
</tr>
<tr>
<td>Maximum cancel attempt time (sec)</td>
<td>The maximum amount of time that the Query Engine waits for a successful attempt to cancel a query.</td>
<td>388</td>
</tr>
<tr>
<td>Maximum query execution time (sec)</td>
<td>The maximum amount of time that a single pass of SQL may execute on the data warehouse.</td>
<td>388</td>
</tr>
<tr>
<td>Maximum connection attempt time (sec)</td>
<td>The maximum amount of time that Intelligence Server waits to connect to the data warehouse.</td>
<td>389</td>
</tr>
</tbody>
</table>

**Database Connection Caching**

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection lifetime (sec)</td>
<td>The amount of time that an active database connection thread remains open and cached on Intelligence Server.</td>
<td>390</td>
</tr>
<tr>
<td>Connection idle timeout (sec)</td>
<td>The amount of time that an inactive database connection thread remains cached until it is terminated.</td>
<td>390</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Governor</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate row limit</td>
<td>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</td>
<td>397</td>
</tr>
<tr>
<td>Results Set Row Limit</td>
<td>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.</td>
<td>395</td>
</tr>
</tbody>
</table>
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 418*)
- How Narrowcast Server connects to Intelligence Server (*How Narrowcast Server connects to Intelligence Server, page 419*)

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 MicroStrategy Desktop 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 Desktop 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 Desktop.

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.
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 422
- The clustered architecture, page 425
- Prerequisites for clustering Intelligence Servers, page 431
- Clustering Intelligence Servers, page 435
- Managing your clustered system, page 445
- Connecting MicroStrategy Web to a cluster, page 453
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 period of 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 435.

• 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. See the documentation for your third-party clustering software for details to implement this clustering method.

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 453.

Benefits of clustering

Clustering Intelligence Servers provides the following benefits:

• Increased resource availability: If one MicroStrategy 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 in the event of a server’s failure.

• Strategic resource usage: You can distribute projects across nodes in whatever configuration you prefer. This reduces overhead since 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 423)

• Load balancing (see Load balancing, page 423)

• Project distribution and project failover (see Project distribution and project failover, page 424)

Failover support

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 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 MicroStrategy Intelligence Servers, so that no single machine is overwhelmed. This strategy is particularly 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. Based on 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. 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 not only provides you with flexibility in using your resources, but can also provide better scalability and performance due to less overhead, since all servers in a cluster do not need to be running all projects.

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.
The clustered architecture

The diagram below shows report distribution in a four-tier clustered environment. The clustered Intelligence Servers are shown in gray.

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 Web clients among Web Servers.

3. The MicroStrategy Web product load balancers on each Web Server collect load information from each cluster node, and then connect the users to the nodes that carry the lightest loads and run the project that
the user logged into. 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.

Query flow in a clustered environment is identical to a standard query flow in an unclustered environment (see *Processing jobs, page 38*), 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 456*. For an introduction to Intelligent Cubes, see *Chapter 10, 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 488*.

**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 427*)

- Result caches and Intelligent Cubes (for details, see *Sharing result caches and Intelligent Cubes in a cluster, page 427*)

- History Lists (for details, see *Synchronizing History Lists, page 431*)

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 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 machine that is running Intelligence Server. For an overview of Intelligent Cubes, see *Chapter 10, Managing Intelligent Cubes*, or see the *MicroStrategy OLAP Services Guide*. For an overview of result caches, see *Result caches, page 456*.

In a clustered environment, each node within 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 given 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.
- Whenever 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 429*.

  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 430*.

  For caches located on Windows machines, and on UNIX/Linux machines using Samba, set the path to `\\<machine name>\<shared cache folder name>`.

  For caches located on UNIX/Linux machines, set the path to `//<SharedLocation>/<CacheFolder>`.
The following table summarizes the pros and cons of the result cache configurations:

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| **Local caching** | • Allows faster read and write operations for cache files created by 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 network account are accessing only one machine for files. | • All cache operations are required to go over the network if shared location is not located on one of the Intelligence Server machines.  
• Requires additional hardware if shared location is not located 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 435*.

**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.
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.

**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 Desktop. *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 488*. For steps to set up History List sharing in a file-based system, see *Configuring History List sharing using multiple local cache files, page 438*.

**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 MicroStrategy Intelligence Server. For more information on using License Manager, see *Chapter 4, 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.

- 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 Desktop through the Intelligence Server Configuration Editor and the Project Configuration Editor. For further instructions, see the Desktop Help. (From within Desktop, press F1.)

- MicroStrategy Desktop must be installed on a Windows machine to administer the cluster. This version of Desktop must be the same as the version of Intelligence Servers. For example, if the Intelligence Servers are running MicroStrategy version 9.0.1, Desktop must also be version 9.0.1.

- You must have access to the Cluster view of the System Administration monitor in Desktop. 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 445*.

- The computers that will be clustered must have the same intra-cluster communication settings. To configure these settings, on each Intelligence
Server machine, in Desktop, right-click on the project source and select **Configure MicroStrategy Intelligence Server**. The Intelligence Server Configuration Editor opens. Under the **Server definition** category, select **General**. For further instructions, see the Desktop 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 427*.

**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 of 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.

- 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 only needs to be set up 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:

  \[ \text{ln } -s \text{ OLDNAME NEWNAME} \]

  where:

  \text{OLDNAME} is the target of the link, usually a path name.

  \text{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 431.

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 between them. For instructions, see Configuring caches in a cluster, page 435.

3. Join nodes.
   You create a cluster by joining Intelligence Servers that have been synchronized. For instructions, see Joining the nodes in a cluster, page 441.

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 between nodes. For instructions on how to perform various tests, see Verifying the clustered system is working, page 442.

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, including instructions, see Distributing projects across nodes in a cluster, page 443.

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 429.
• **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 430.*

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 436*
- *Configuring caches in a cluster on UNIX/Linux, page 438*

**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 only need to configure the cache location in Intelligence Server 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 427.*

---

**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\MicroStrategy\ Intelligence Server\Caches\Server Definition`
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 Shared 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**.

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 MicroStrategy 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 436.* The History List messages are stored in the History folder. (To locate this folder, in the Intelligence Server Configuration Editor, select **Governing**, then select **History settings**.)

The History List location is `.\Inbox\ServerDefinition`, where **Server Definition** is the name of the folder containing the History Lists. This folder must be shared with the share name “ClusterInbox”, since 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 426.*
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 426*.

---

**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 Desktop, create project sources pointing to UNIX1 and UNIX2.
3. Connect to UNIX1 using Desktop.
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 located 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 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:

- 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 Desktop, create project sources pointing to UNIX1 and UNIX2.

3. Connect to UNIX1 using Desktop.

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:

   ```bash
   mkdir $MSTR_<HOME_PATH>/ClusterCaches
   mkdir $MSTR_<HOME_PATH>/ClusterInbox
   ```

12 Mount the folders from UNIX2 on UNIX1. For example:

   ```bash
   mkdir /UNIX2
   mount UNIX2:/Build/BIN/SunOD /UNIX2
   ```

**To set up the UNIX2 machine**

13 Create the folders for caches:

   ```bash
   mkdir $MSTR_HOME_PATH/ClusterCaches
   mkdir $MSTR_HOME_PATH/ClusterInbox
   ```

14 Mount the folders from UNIX2 on UNIX1. For example:

   ```bash
   mkdir /UNIX1
   mount UNIX1:/Build/BIN/SunOD /UNIX1
   ```

15 Restart both Intelligence Servers.

**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 Desktop, 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 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 wish to add 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 Desktop**

**Verify the Cluster view**

1 Connect to one Intelligence Server in the cluster and ensure that the Cluster view in Desktop (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.
Without logging out that user, log on to a different node with the same user name.

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 Desktop*.

**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 a given Intelligence Server. This provides you with flexibility in using your resources, as well as better scalability and performance.

To distribute projects across the cluster, you manually assign the projects to specific nodes within 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 will result 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 *Chapter 14, Analyzing System Usage with Enterprise Manager*.

**To distribute projects across nodes in a cluster**

1. In MicroStrategy Desktop, 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 given project on a given 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.

   - 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 will result in a loss of session statistics for any Intelligence Server onto which the project is not loaded at startup. For more information about single instance session logging, see *Chapter 14, Analyzing System Usage with Enterprise Manager*. 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 place an individual check mark for a given 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 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 accomplish this. See the *MicroStrategy Installation and Configuration Guide* for details.

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.

---

**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 446*
- *Project failover and latency, page 448*
- *Shutting down a node, page 451*
- *Maintaining result caches in a clustered environment, page 452*
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 only need to perform maintenance on a single 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 31.

To manage the projects and nodes in a cluster

1. In Desktop, 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.
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 currently executing and queued jobs finish executing, and any newly submitted jobs are rejected.
   • Execution Idle (Execution Idle for All Jobs): all currently 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 currently 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 currently executing and queued jobs are cancelled, and any newly submitted jobs are rejected.
- Partial Idle (Request Idle and Execution Idle for Warehouse jobs): all currently executing and queued jobs that do not submit SQL against the data warehouse are cancelled, 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.

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 within 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 server 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 given 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 only take effect 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 a different machine
- After the failed machine is recovered, but before its original projects are reloaded

**To set project failover latency**

1. In MicroStrategy Desktop, from the Administration menu, select Projects, then select Select Projects. The Intelligence Server Configuration Editor opens, with the Projects: General subcategory displayed.

2. Select the Advanced subcategory.

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 particularly 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.

**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 only takes effect 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 can be a good idea 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 back 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 can be a good idea if your projects are large and you want to be sure your recovered server stays online for a specific period of time 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 can be a good idea 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.
Shutting down a node

A node may 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 due to a forceful shutdown, its cache resources are still valid to other nodes in the cluster and will be 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**

**Desktop**

Client connections that are not cluster-aware, such as Desktop, 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 Desktop clients connected to that Intelligence Server receive an error message notifying them of the lost connection, regardless of whether or not that Intelligence Server was in a cluster.
MicroStrategy Web

If a cluster node shuts down while there are MicroStrategy Web users 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 will no longer have access to the 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 to perform this configuration, see the Desktop 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.

Nodes that are still in the cluster but not available are listed in the Cluster Monitor with a status of Stopped.

Maintaining result caches in a clustered environment

Proper maintenance of result caches is important in any MicroStrategy system. For detailed information on caches and cache management, including recommended best practices, see Result caches, page 456.
When maintaining result caches in a clustered environment, be aware of the following:

- You can only manage the caches on a given node 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 there is only one valid copy of a cache at any time for a report on all nodes in the cluster. For more information about invalidating caches, see *Invalidating result caches*, page 472.

- The Cache Monitor’s hit count number on a given machine only reflects the number of cache hits the given machine initiated on any cache within the cluster. If a different machine within the cluster hits a cache on the local machine, that hit will not be counted on the local machine’s hit count. For more information about the Cache Monitor, see *Monitoring result caches*, page 466.

For example, ServerA and ServerB are clustered together, and the cluster is configured to use local caching (see *Local caching*, page 429). 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 only displays activity initiated by ServerA.

- MicroStrategy recommends that in a clustered environment, you set the cache backup frequency to 0 (zero) to ensure that History List messages are synchronized correctly. This setting is found in the Intelligence Server Configuration Editor, in the Server Definition: Advanced category. For more information about this setting, see *Backup Frequency (minutes)*, page 477.

### 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 labelled 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 specific 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 will depend 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 435*.

**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**.
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 456*.

  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 OLAP Services 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 488*. 
• **Element caches**: Most-recently used lookup table elements that are stored in memory on the Intelligence Server or MicroStrategy Desktop machines so they can be retrieved more quickly. For more information on these, see *Element caches, page 504.*

• **Object caches**: Most-recently used metadata objects that are stored in memory on the Intelligence Server and MicroStrategy Desktop machines so they can be retrieved more quickly. For more information on these, see *Object caches, page 517.*

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 7, 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 Desktop.

Report caches can only be created or used for a project 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 only be created or used for a project 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 only created or used when a document is executed in MicroStrategy Web. Document caches are not created or used when a document is executed from Desktop.

By default, result caching is enabled at the project level. It can also be set on a per-report and per-document basis. 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 487.

A result cache is created when you do any of the following:

• In MicroStrategy Web or Desktop, execute a saved report or document containing only static objects.

• In MicroStrategy Web or Desktop, 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, Desktop, Mobile, Distribution Services, or Narrowcast Server. For more information about scheduling reports, see Scheduling reports and documents: Subscriptions, page 553.

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 466.

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 778 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.”

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 459*
- *Types of result caches, page 460*
- *Location of result caches, page 461*
- *Cache matching algorithm, page 463*
- *Disabling result caching, page 466*
- *Monitoring result caches, page 466*
- *Managing result caches, page 470*
- *Configuring result cache settings, page 476*
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:

• 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 Chapter 14, Analyzing System Usage with Enterprise Manager. For information about enabling caching on a report-by-report basis, see Result cache settings at the report level, page 487.

• 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 OLAP Services Guide. For information about purchasing OLAP Services, contact your MicroStrategy account representative.

• If results are cached by user ID (see Create caches per user, page 481), 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 488.

• 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 476.

• 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 in a clustered environment, page 452.
Types of result caches

The following types of result caches are created by Intelligence Server:

- Matching caches, page 460
- History caches, page 460
- Matching-History caches, page 461
- XML caches, page 461

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, while a History cache holds the data for a History List message and is only accessed when that History List message is retrieved.

For more information about History Lists, see *Saving report results: History List, page 488.*

**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 Web, page 78.*

**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.

The amount of memory available to store result caches is limited by the Memory Storage settings. For information, see *Maximum RAM usage, page 483.*
Result cache files

By default, result cache files are stored in the directory where Intelligence Server is installed, in \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 463*). 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 476*).

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

MicroStrategy Intelligence Server creates two types of index files to identify and locate document caches:

- **RWDPool.idx**: An index file that contains a list of all Matching caches and pointers to the caches’ locations.
- **RWDLkUp.idx**: 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 463). 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 476.)

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.
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 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 88).

- **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 123.

- **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 487.

To disable result caching

1. Open the Project Configuration Editor for the project.
2. Select the Caching: Result caches: Creation category.
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.

Monitoring result caches

You use the Cache Monitor in MicroStrategy Desktop 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 Diagnostic and Performance Logging Tool, page 469),
and Command Manager to automatically update information about result caches (see Command Manager, page 470).

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 Desktop, 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.
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 470.

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 778.

### Cache statuses

A result cache’s status is displayed in the Report Cache Monitor using one or more of the following letters:

<table>
<thead>
<tr>
<th>Status</th>
<th>Stands for</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Ready</td>
<td>The cache is valid and ready to be used.</td>
</tr>
<tr>
<td>P</td>
<td>Processing</td>
<td>The cache is currently being updated.</td>
</tr>
<tr>
<td>I</td>
<td>Invalid</td>
<td>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 <em>Invalidating result caches</em>, page 472.</td>
</tr>
<tr>
<td>E</td>
<td>Expired</td>
<td>The cache has been invalidated because its lifetime has elapsed. For information about expired caches, see <em>Expanding result caches</em>, page 475.</td>
</tr>
<tr>
<td>L</td>
<td>Loaded</td>
<td>The cache is loaded into Intelligence Server memory.</td>
</tr>
<tr>
<td>U</td>
<td>Updated</td>
<td>The cache file has been updated.</td>
</tr>
<tr>
<td>D</td>
<td>Dirty</td>
<td>The cache has been updated in Intelligence Server memory since the last time it was saved to disk.</td>
</tr>
<tr>
<td>F</td>
<td>Filed</td>
<td>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 <em>Unloading and loading result caches to disk</em>, page 471.</td>
</tr>
</tbody>
</table>
Cache types

Result caches can be of the following types:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching</td>
<td>The cache is valid and available for use. Note: All document caches are Matching caches.</td>
</tr>
<tr>
<td>History</td>
<td>The cache referenced in at least one History List message.</td>
</tr>
<tr>
<td>Matching-History</td>
<td>The cache is valid and available for use, and also referenced in at least one History List message.</td>
</tr>
<tr>
<td>XML</td>
<td>(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.</td>
</tr>
</tbody>
</table>

For more information about each type of cache, see *Types of result caches*, page 460.

Diagnostic 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 740.

To enable diagnostic tracing of result caches

1. Open the MicroStrategy Diagnostics and Performance Logging Tool. (From the Windows Start menu, point to Programs, then MicroStrategy, then 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.

**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 specified project.

- **LIST [ALL] PROPERTIES FOR REPORT CACHE "<cache_name>" IN PROJECT "<project_name>"** lists information about a specific report cache.

These scripts are located at **C:\Program Files\MicroStrategy\Command Manager\Outlines\Cache.Outlines**.

For more information about Command Manager, see **Chapter 12, 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 471**
- **Unloading and loading result caches to disk, page 471**
- **Invalidating result caches, page 472**
- **Deleting result caches, page 474**
- **Purging all result caches in a project, page 474**
- **Expiring result caches, page 475**

### 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 553*.

### 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 based on the Backup frequency setting (see *Backup Frequency (minutes), page 477*). Caches are always saved to disk regardless of whether they are loaded or unloaded; unloading or loading a cache only affects 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 based on outdated cached data. Examples of when the data might 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 477.

You can invalidate caches manually or by scheduling the invalidation process.
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 560.*

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 located at `C:\Program Files\MicroStrategy\Command Manager\Outlines\Cache_Outlines\Invalidate_Report_Cache_Outline`. For more information about Command Manager, see *Chapter 12, 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 Desktop, 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 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 caches 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 477*.

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 472*.

**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 516 and Deleting object caches, page 520.

To purge all result caches in a project

1 In Desktop, right-click the project and select Project Configuration Editor. The Project Configuration Editor opens.

2 Expand Caching, then Result Caches, and then select Maintenance.

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 472.

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 only affects 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 476]*)
- At the project level (see *[Result cache settings at the project level, page 478]*)
- At the individual report/document level (see *[Result cache settings at the report level, page 487]*)

Each is discussed in detail below.

⚠️ Changes to any of the caching settings are effective 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 477**
- **Cache lookup cleanup frequency (sec), page 477**

You can also configure these settings using the Command Manager script, `Alter_Server_Config_Outline.otl`, located at C:\Program Files\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 542.

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 479
  - Enable document output caching in selected formats, page 479
  - Enable prompted report and document caching, page 479
  - Record prompt answers for cache monitoring, page 480
  - Enable non-prompted report and document caching, page 480
  - Enable XML caching for reports, page 480
  - Create caches per user, page 481
  - Create caches per database login, page 481
  - Create caches per database connection, page 481

- In the Result caches (storage) subcategory
  - Cache file directory, page 482
  - Cache encryption level on disk, page 483
  - Maximum RAM usage, page 483 (separate settings for report and document caches)
  - Maximum number of caches, page 484 (separate settings for report and document caches)
  - RAM swap multiplier, page 484
  - Maximum RAM for cache index %, page 485
  - Load caches on startup, page 485

- In the Result caches (maintenance) subcategory
  - Never expire caches, page 485
  - Cache duration (in hours), page 485
  - Cache expiration and dynamic dates, page 486
  - Purging all result caches in a project, page 474
• In the Subscription Execution subcategory
  - Cache usage defaults for subscriptions, page 486

To locate these settings, right-click the project and select Project Configuration. Then, in the Project Configuration Editor, expand Caching, and then select Result caching.

You can also configure these settings using Command Manager scripts located at C:\Program Files\MicroStrategy\Command Manager\Outlines\Cache_Outlines.

**Enable report server caching**

Result caches can only be created or used for a project if the Enable report server caching check box is selected in the Project Configuration Editor in the Caching: Result Caches (Creation) category.

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 456.

**Enable document output caching in selected formats**

Document caches can only be created or used for a project 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 any or all of PDF, Excel, HTML, and XML/Flash.

Document caches are only created or used when a document is executed from MicroStrategy Web. They are not created or used in Desktop.

**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, select its check box in the Project Configuration Editor under the Caching: Result Caches (Creation) category.

**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 461*.

This option is enabled by default. To disable it, select 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 463.

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 488*.

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 463*.

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.

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 463*.

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.

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. ClusterCaches 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 located 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 MicroStrategy 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 435.*
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 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 461*.

**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 MicroStrategy 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.

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 483*) 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%.

**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 483*). 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 *Expanding result caches, page 475*.

**Cache duration (in hours)**

All caches that have existed for longer than the **Cache Duration (in hours)** are automatically expired. This duration is set to 24 hours by default. You
can change the duration in the 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 475.

**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 553.

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.

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 489
• Configuring History List data storage, page 492
• History List backup frequency, page 496
• History Lists in a clustered environment, page 496
• Accessing History Lists, page 496
• Archiving History List messages, page 498
• Managing History Lists, page 500
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 two storage options, see Configuring History List data storage, page 492.

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 in the 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 Desktop:** 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.
    
    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 Desktop:** 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 Desktop**: 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 Desktop, 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 Web**: Select **History List** from the Project Preferences, and then select **Automatically for Add reports and documents to my History List**.
  
  - **From Desktop**: Select **Desktop 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 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 Desktop: 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 MicroStrategy Desktop

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 only deleted once all messages pointing to it have been deleted.

If you are using a database-based History List repository, 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 460*. For more information about storing History List data, see *Configuring History List data storage, page 492*.

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 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 two different ways that the History List repository can be configured to store data for the History List. It can either be stored in a database, or in a file on the Intelligence Server machine.

**Configuring Intelligence Server to use a database-based History List repository**

The caches associated with History Lists can be stored in a database. Storing the History List cache in a database reduces the load on the machine that hosts Intelligence Server. If you are using a database-based History List repository, the report caches that are associated with a History List message are also stored in the database. For more information about report caches and History List messages, see *History Lists in a clustered environment, page 496*.

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. If you want to revert back to a file-based repository, you must change the server definition.

**Prerequisites**

- The storage location for the History List data (the History List repository) has been created in the database.
A database instance has been created that points to the History List repository in the database.

For information about creating the History List repository in the database and about creating database instances, see the MicroStrategy Installation and Configuration Guide.

To configure Intelligence Server to use a database-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 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. If you want to revert back to a file-based repository, you must change the server definition.

5. Click Yes. The warning message closes.

6. From the Database Instance menu, select the database instance that points to the History List repository in the database.

7. Click OK. The Intelligence Server Configuration Editor closes.

8. Restart Intelligence Server for the changes to take effect.
To confirm that the History List repository has been configured correctly

9 Log in to the project source as a user with administrative privileges.

10 From the Administration menu, select Server, then Configure MicroStrategy Intelligence Server. The Intelligence Server Configuration Editor opens.

11 On the left, expand History Settings and select General. If you have configured Intelligence Server properly, following message is displayed in the Repository Type area of the Intelligence Server Configuration Editor:

![Server is currently connected to the History List Repository.

Configuring Intelligence Server to use a file-based History List repository

It is also possible to store the History List cached data in a file on the machine that hosts Intelligence Server. The default location of this file is relative to the installation path of Intelligence Server:

.\Inbox\<Server definition name>

For example, C:\Program Files\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 located on Windows machines, and on UNIX/Linux machines using Samba, set the path to `\<machine name>\<shared directory name>`. For caches located 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 MicroStrategy 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.

### Configuring 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.
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 477.

In a History List shared by a cluster of Intelligence Servers, the servers must be synchronized to allow sharing History List information, and the backup frequency must be set to 0. This means that cache and History List messages are backed up immediately after they are created. For more information about how History List works in clustered environments, see Synchronizing History Lists, page 431.

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 431.

Accessing History Lists

History Lists can be accessed from both Web and Desktop. 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 500.

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 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 MicroStrategy Desktop**

In MicroStrategy Desktop, 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 or not 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.

**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 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 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 report will overwrite older versions of itself**.
To archive History List messages in Desktop

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:
   - 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 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

When users manage History Lists, they are managing the History caches at the same time. For information on the relationship between the History caches and History Lists, see *History caches, page 460*.

Although you can set the number of History List messages retained in the History List database or folder to a relatively high number (the maximum is 10,000), keep in mind that if the list gets too big, you run the risk of wasting resources. Therefore, MicroStrategy recommends that you educate users to make efficient use of the History List feature by keeping only needed messages and deleting unneeded ones in a timely fashion.

While users can do their part to maintain the size of the History List, an administrator can control the size of the History List and thus control resource usage through the following settings:

• 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 501*.

• You can also delete History List messages according to a schedule. For more details, including instructions, see *Scheduling History List message deletion, page 501*.

• 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 502*.

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 503*. 
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 460*.

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 Desktop, 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 Desktop, 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 ... (browse) to select a user/group for which the History List messages will be deleted.

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 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 Desktop, 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 Back up 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 History List repository, page 492.

To monitor the History List messages

1  In Desktop, 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:

<table>
<thead>
<tr>
<th>User</th>
<th>Project</th>
<th>Message Name</th>
<th>Message Text</th>
<th>Creation Time</th>
<th>Status</th>
<th>Read Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guest User</td>
<td>MicroStrategy Tutorial</td>
<td>Employee Profitability Analysis</td>
<td>User request</td>
<td>1/26/2009 4:21:06 PM Ready</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Administrator</td>
<td>MicroStrategy Tutorial</td>
<td>Revenue Status and Trend Analysis</td>
<td>User request</td>
<td>1/26/2009 4:11:21 PM Ready</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Administrator</td>
<td>MicroStrategy Tutorial</td>
<td>Revenue over Time</td>
<td>User request</td>
<td>1/26/2009 4:11:18 PM Ready</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Administrator</td>
<td>MicroStrategy Tutorial</td>
<td>Revenue by Region</td>
<td>User request</td>
<td>1/26/2009 4:11:17 PM Ready</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Administrator</td>
<td>MicroStrategy Tutorial</td>
<td>Revenue and Profit Trends by...</td>
<td>User request</td>
<td>1/26/2009 4:11:14 PM Ready</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Administrator</td>
<td>MicroStrategy Tutorial</td>
<td>Electronics Sales over Time</td>
<td>User request</td>
<td>1/26/2009 4:11:11 PM Ready</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Administrator</td>
<td>MicroStrategy Tutorial</td>
<td>Electronics Revenue by...</td>
<td>User request</td>
<td>1/26/2009 4:11:11 PM Ready</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

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 MicroStrategy Desktop machines so they can be retrieved more quickly. They are created when users:

- Browse attribute elements in MicroStrategy Desktop 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 506
- **Location of element caches**, page 506
- **Cache matching algorithm**, page 507
- **Enabling or disabling element caching**, page 507
- **Limiting the number of elements displayed and cached at a time**, page 508
- **Caching algorithm**, page 511
- **Limiting the amount of memory available for element caches**, page 512
- **Limiting which attribute elements a user can see**, page 513
- **Limiting element caches by database connection**, page 514
- **Limiting element caches by database login**, page 515
- **Deleting all element caches**, page 516
- **Summary table of element cache settings**, page 516
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: Caching - 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.

- **Element Incremental Fetch Size**: The maximum number of elements for display in the interface per element request. On Desktop, 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 MicroStrategy Desktop machines.

When a Desktop user triggers an element request, the cache within the Desktop 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 Desktop, 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 Desktop 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 than once a day, it may be desirable to disable element caching.

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 MicroStrategy Desktop

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 Desktop 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 Desktop, 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 Attribute element browsing setting in the Caching: Elements category. The default value is 1,000 for Desktop 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 hit the next option four times to introduce another SELECT pass, which will retrieve 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% of the element cache on any single cache request. For example, if 200 elements use 20% of the cache pool, Intelligence Server only caches 100 elements, which is 10% of the available memory for element caches.

The number of elements retrieved per element cache can be set for Desktop 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 Desktop users)

1 Open the Project Configuration Editor and select the Project definition: Advanced category.

2 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 From the MicroStrategy Web or Web Universal interface, select Preferences.

2 Select Project Defaults in the Preferences Level category.

3 Select General in the Preferences category.

4 Type the limit for the Maximum number of attribute elements per block setting in the Incremental Fetch subcategory.
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 will only retrieve 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 only
counted 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 of 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 MicroStrategy Desktop 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:

- 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  Open the Project Configuration Editor and select the Caching: Auxiliary Caches (Elements).

2  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.

3  Specify the amount of RAM (in megabytes) in the **Client: 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 MicroStrategy Desktop

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 **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 is 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 95.*

---

**To limit which attribute elements users can see per project**

1. In the Project Configuration Editor, select the **Project definition:** Advanced category.

2. 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**).

### 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 the Project Configuration Editor, select the Caching: Auxiliary Caches (Elements) category.

2. 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 88.

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.

To limit element caches by database login

1. In the Project Configuration Editor, select the Caching: Auxiliary Caches (Elements) category.
2 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 Desktop and Intelligence Server machines. This does not delete element caches on other Desktop machines. You cannot delete only certain caches; all of them are 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 474* and *Deleting object caches, page 520*.

---

**To delete all element caches for a project**

1 In the Project Configuration Editor, select the Caching: Auxiliary Caches (Elements) category.

2 Click **Purge Now**.

All element caches are automatically purged whenever schema is updated.

### 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.

<table>
<thead>
<tr>
<th>Setting</th>
<th>For information...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of elements to display</td>
<td>see Limiting the number of elements displayed and cached at a time, page 508</td>
</tr>
<tr>
<td>Attribute element number count method</td>
<td>see Optimizing element requests, page 510</td>
</tr>
<tr>
<td>Element cache - Max RAM usage (MBytes) Project</td>
<td>see Limiting the amount of memory available for element caches, page 512</td>
</tr>
<tr>
<td>Element cache - Max RAM usage (MBytes) Desktop</td>
<td>see Limiting the amount of memory available for element caches, page 512</td>
</tr>
<tr>
<td>Apply security filter to element browsing</td>
<td>see Limiting which attribute elements a user can see, page 513</td>
</tr>
<tr>
<td>Create caches per connection map</td>
<td>see Limiting element caches by database connection, page 514</td>
</tr>
<tr>
<td>Create caches per passsthrough login</td>
<td>see Limiting element caches by database login, page 515</td>
</tr>
<tr>
<td>Purge element caches</td>
<td>see Deleting all element caches, page 516</td>
</tr>
</tbody>
</table>

**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 MicroStrategy Desktop and MicroStrategy 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 MicroStrategy Desktop or MicroStrategy 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 MicroStrategy Intelligence Server and also on the MicroStrategy Desktop of the user who submitted the request. As with element caching, any time the object definition can be returned from memory in either the Desktop or Intelligence Server machine, it is faster than retrieving it from the metadata database.
So when a Desktop user triggers an object request, the cache within the Desktop 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 518**
- **Enabling or disabling object caching, page 518**
- **Limiting the amount of memory available for object caches, page 519**
- **Deleting object caches, page 520**
- **Summary table of object caching settings, page 521**

### 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 MicroStrategy Desktop 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
- 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 10 MB. Intelligence Server estimates that each object consumes 5 KB of the cache pool, so by default it caches 2,048 objects in server memory and 2,048 objects in client memory.

---

**To set the RAM available for object caches for a project**

1. Open the Project Configuration Editor and select the **Caching: Auxiliary Caches (Objects)** category.
2. Specify the RAM (in megabytes) in the **Server: Maximum RAM usage (MBytes)** box.
3 Specify the RAM (in megabytes) in the **Client: Maximum RAM usage (MBytes)** box.  

The new settings take effect only after Intelligence Server is restarted.

---

**To set the RAM available for object caches for a MicroStrategy Desktop 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 Desktop and Intelligence Server machines. However, this does not delete object caches on other Desktop machines. You cannot select to delete only certain object caches; all of them are 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 474* and *Deleting all element caches, page 516*.

---

**To delete all object caches for a project**

1 Open the Project Configuration Editor and select the **Caching: Auxiliary Caches (Objects)** category.
2 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 Desktop, point to **Server**, and then select **Purge Server Object Caches**.

You cannot automatically schedule the purging of server object caches from within MicroStrategy Desktop. 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 12, Automating Administrative Tasks with Command Manager*.

---

**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.

<table>
<thead>
<tr>
<th>Setting</th>
<th>For information...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object cache - Max RAM usage (MBytes) Project level</td>
<td>See <em>Limiting the amount of memory available for object caches, page 519</em></td>
</tr>
<tr>
<td>Object cache - Max RAM usage (MBytes) MicroStrategy Desktop</td>
<td>See <em>Limiting the amount of memory available for object caches, page 519</em></td>
</tr>
<tr>
<td>Purge object caches</td>
<td>See <em>Deleting object caches, page 520</em></td>
</tr>
</tbody>
</table>
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 add-on to Intelligence Server. For detailed information about Intelligent Cubes, see the MicroStrategy OLAP Services Guide. For information about purchasing OLAP Services, contact your account executive.

This chapter contains the following information:

- Managing Intelligent Cubes: Intelligent Cube Monitor, page 524
- Governing Intelligent Cube memory usage, loading, and storage, page 532
- Supporting connection mappings in Intelligent Cubes, page 543
Managing Intelligent Cubes: Intelligent Cube Monitor

You must create Intelligent Cubes before they can be published. For information on creating Intelligent Cubes, see the OLAP Services 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 778.

To view the available Intelligent Cubes

1. In Desktop, 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 526.
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.

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 526*.
- **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 updated.
- **Historic Hit Count**: The total number of times the Intelligent Cube has been used by reports. Unpublishing and republishing an Intelligent Cube creates a new Intelligent Cube that resets this value to zero.
- **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 rows 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:

<table>
<thead>
<tr>
<th>Action</th>
<th>Required Status to Perform Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate</td>
<td>Filed, but not Active</td>
<td>Loads a previously deactivated Intelligent Cube as an accessible set of data for multiple reports.</td>
</tr>
<tr>
<td>Deactivate</td>
<td>Active</td>
<td>Removes an Intelligent Cube instance from Intelligence Server memory, but saves it to secondary storage, such as a hard disk.</td>
</tr>
</tbody>
</table>
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 542), 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 541*.

**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 529*
- *Loading Intelligent Cubes when Intelligence Server starts, page 540*

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 534*.

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 7, Tuning your System for Best Performance*.

**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 Desktop, 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. Select the **Intelligent Cubes** category.

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.
To define whether publishing Intelligent Cubes loads them into Intelligence Server memory, at the Intelligent Cube level

1. In Desktop, 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 negatively affect Intelligence Server’s 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 528.*

- **Governing Intelligent Cube memory usage, page 532**
- **Loading Intelligent Cubes when Intelligence Server starts, page 540**
- **Storing Intelligent Cubes in secondary storage, page 541**

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 potentially 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 an memory limit on Intelligent Cubes to be 512MB. Currently 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 factors listed below:

- 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 System Administration Guide.

- 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 can be unloaded until they are required by reports (see Loading and unloading Intelligent Cubes, page 528).

- 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 potentially 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 act of publishing an Intelligent Cube can require memory resources in the area of two to four times greater than the size of an Intelligent Cube. This can affect performance of your Intelligence Server as well as 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* 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, as 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**

The act of 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 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 529*. Loading and unloading Intelligent Cubes is described in *Loading and unloading Intelligent Cubes, page 528*.
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 OLAP Services Guide.

To determine memory limits for Intelligent Cubes, you should review the considerations listed in Determining memory limits for Intelligent Cubes, page 532. 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.

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 define. 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 Desktop, 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 From the **Categories** list, expand **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 number of Intelligent Cube that can be stored in Intelligence Server memory for a project at one time. 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 Desktop, 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 Under the **Intelligent Cubes** category, 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% defines that a 100 MB Intelligent Cube will only be allowed to grow to 150 MB due to 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**: 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.

**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 Desktop, 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 From the Categories list, expand **Governing Rules**, and choose **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 is 30 MB. The minimum value is 1 MB, and the maximum value is 500 MB.

- **Maximum quota per user (MB)**: Defines the maximum total memory available for all Intelligent Cubes that the user can create using the Import Data feature. The default is 100 MB. Type -1 if you want the total size to be unlimited.
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:

<table>
<thead>
<tr>
<th>Method</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| 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 534.

To enable or disable loading Intelligent Cubes when Intelligence Server starts

1 In Desktop, 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 From the Categories list, expand Cubes, and then select General.

4 Select or clear the Load 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.

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 Desktop, 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 From the **Categories** list, expand **Cubes**, and then select **General**.

4 In the **Cube file directory area**, click the ... 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* 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 Desktop, 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 477**.

5 Click **OK** to save your changes and close the MicroStrategy Intelligence Server Configuration Editor.

6 For your changes to be applied, you must restart Intelligence Server.

---

### 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 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 J Doe who is assigned to connection mapping A publishes Intelligent Cube X. The Intelligent Cube is published using connection mapping X. User F Johnson who is assigned connection mapping B cannot create and execute a report connected to Intelligent Cube X. To allow F Johnson 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 Desktop, 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. From the **Categories** list, 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.
SCHEDULING JOBS AND ADMINISTRATIVE TASKS

Introduction

Scheduling is a feature of MicroStrategy 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 since scheduling reports, in conjunction with other features such as caching and clustering, can improve the overall performance of the system. Certain administration tasks can also be scheduled.

Intelligence Server executes a task in exactly the same manner whether it is scheduled or not. All governing parameters and error conditions apply to scheduled tasks in the same way 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 topics concerning scheduling:

• Best practices for scheduling jobs and administrative tasks, page 546
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 a large number of reports or tasks that need to be executed on the same time-based schedule, consider creating several similar schedules that trigger fifteen minutes apart. For example, one schedule triggers at 8 AM every Monday morning, 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 69.

- 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 different reports at the same time.

- If you need to temporarily disable a schedule, you can do so by setting its start date for some time in the future. The schedule will 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 there are many subscriptions listed in the Subscription Manager, you can filter the list of subscriptions so that you only see the relevant
subscriptions. For more information, including instructions, see the Subscription Manager Help. (In Desktop, press F1.)

- 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 will be unable to run the report successfully since the prompt cannot be resolved, and the subscription will be automatically invalidated and removed from the system.

- When a scheduled report/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 simply opens the message to retrieve the results. If the request was not completed successfully, the user can view details of the error message. These messages are available in the History List folder. For more information about History Lists, see Saving report results: History List, page 488.

- 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 category, enable the administrator notification settings for failed deliveries.

Creating and managing schedules

A schedule is a MicroStrategy object that contains information specifying when a task is to be executed. A single schedule can control several different tasks. Schedules are stored at the project source level, and are thus available to all projects within the project source.

Intelligence Server supports two types of schedules:

- Time-triggered schedules execute at a specific date and time, or at a specific recurring date and time. For details, see Time-triggered schedules, page 548.

- Event-triggered schedules execute when the event associated with them is triggered. For details, see Event-triggered schedules, page 548.
Time-triggered schedules

With a time-triggered schedule, you define a specific date and time at which the scheduled task is to be run. For example, you can execute a particular 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 Desktop 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 the primary node of the cluster only.

Event-triggered schedules

An event-triggered schedule causes tasks to occur when a specific 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 550.

In a clustered environment, administrative tasks associated with event-triggered schedules are executed only on the node of the cluster that triggered the event.

Creating schedules

To create schedules, you need to 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 69*.

To create effective and useful schedules, you need to have a clear understanding of your users’ needs as well as the usage patterns of the overall system. Schedules need to be created in advance before they are linked to any tasks.

### To create a schedule

1. In Desktop, 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 550*. 
To manage your schedules in the Schedule Manager

1. In Desktop, 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 548.
   - To modify a schedule, right-click the schedule and select Edit. The Schedule Wizard opens with the information for that schedule. 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 a given schedule, right-click the schedule and select Search for dependent subscriptions. A list of subscriptions that use that schedule opens.

About events and event-triggered schedules

Subscriptions and tasks based on event-triggered schedules (see Event-triggered schedules, page 548) occur 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 551.

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 only executed 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 563*.

**Creating events**

You can create events in Desktop using the Event Manager.

---

**To create an event in Desktop**

1. In Desktop, 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:

```sql
CREATE EVENT event_name [DESCRIPTION description];
```

By default, this script is located at `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 12, Automating Administrative Tasks with Command Manager*. 

© 2011 MicroStrategy, Inc.
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.

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. For information about obtaining the MicroStrategy SDK, contact your MicroStrategy account representative.

**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 is often not practical for the administrator to be present to trigger event-based schedules.

**To trigger an event manually**

1. In Desktop, 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 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.

Scheduling the execution of reports and documents 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 any additional load on the database system.
- Large, long-running reports and documents can be postponed to a later time when the system load is lighter.

A subscription for a document only creates or updates 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 463.

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. Personalization options vary depending on what a given 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 554
- Creating subscriptions, page 555
- Managing your subscriptions, page 559
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 471. For detailed information about caches, see *Result caches*, page 456.

- **History List** subscriptions create a History List message for the specified report or document. The user can then retrieve the report or document from the History List message. For detailed information about the History List, see *Saving report results: History List*, page 488.

- **Mobile** subscriptions deliver the report or document to a mobile device such as an iPhone or BlackBerry, via MicroStrategy Mobile. These subscriptions are only 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 OLAP Services Guide*.

If you have purchased a Distribution Services license, you can also schedule reports and documents to be emailed to users, saved as Excel or PDF files, or printed, as described below. For instructions on configuring Distribution Services, see *Scheduling deliveries to email, file, and printer using Distribution Services*, page 564.

- **Email** subscriptions deliver a report or document to one or more email addresses.
• **File** subscriptions save the report or document to a disk location on the network.

• **Print** subscriptions automatically print a report or document from a specified printer.

## Creating subscriptions

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 Desktop, or through MicroStrategy Web (see *To subscribe to a report or document in Desktop, page 556* or *To create a subscription in MicroStrategy Web, page 557*).

• You can create multiple subscriptions at one time for a user or user group using the Subscription Wizard in Desktop (see *To create multiple subscriptions at one time in Desktop, page 557*).

• 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 12, Automating Administrative Tasks with Command Manager*, or see the Command Manager Help. (From within Command Manager, select Help.)

## 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 have default prompt answers set. The
following table explains how Intelligence Server resolves the different possible scenarios that can occur for each prompt in a subscribed report.

<table>
<thead>
<tr>
<th>Prompt Required?</th>
<th>Default Answer present?</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>The prompt is ignored since it is not required; the report is executed but it is not filtered by the prompt.</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>The prompt and default answer are ignored since the prompt is not required; the report is executed but it is not filtered by the prompt.</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>The report is not executed. No answer is provided to the required prompt so MicroStrategy cannot complete the report without user interaction.</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>The report is executed; the prompt is answered with the default.</td>
</tr>
</tbody>
</table>

**Prerequisites**

- To create any subscriptions, you must have the Schedule Request privilege.
- To create email, file, or print subscriptions, you must have a Distribution Services license.
- To create mobile subscriptions, you must have a MicroStrategy Mobile license.
- To subscribe other users to a report or document, you must have the Administer Subscriptions privilege (for Desktop) or the Web Subscribe Others privilege (for MicroStrategy Web).

---

**To subscribe to a report or document in Desktop**

- Only History List, cache, Intelligent Cube, and Mobile subscriptions can be created in Desktop.

1. In Desktop, select the report, document, or Intelligent Cube to have 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 554*. 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 Desktop

1 In Desktop, 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 information, including step-by-step 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 button.

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 554. 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 opens. Select the users or groups and click OK. The Recipient Browser closes and the users and groups are added to the recipient list.
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. A History List message is automatically generated whenever a report or document is executed in Web based on a schedule.

### Personalizing email and file subscriptions

If you have the Distribution Services product, you can personalize your email and file subscriptions with macros in the File Name, Subject, Message, or Zip File Name 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 580*.

The table below lists the macros that can be used in email and file subscriptions, and the fields in which they can be used:

<table>
<thead>
<tr>
<th>Text</th>
<th>Macro</th>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date the subscription is sent</td>
<td>{&amp;Date}</td>
<td>Subject, File Name</td>
</tr>
<tr>
<td>Time the subscription is sent</td>
<td>{&amp;Time}</td>
<td>Subject, File Name</td>
</tr>
<tr>
<td>Name of the recipient</td>
<td>{&amp;RecipientName}</td>
<td>Subject, File Name</td>
</tr>
<tr>
<td>User login</td>
<td>{&amp;UserLogin}</td>
<td>All fields</td>
</tr>
<tr>
<td>Name of the subscription</td>
<td>{&amp;Subscription}</td>
<td>All fields</td>
</tr>
<tr>
<td>Project that contains the subscribed report/document</td>
<td>{&amp;Project}</td>
<td>All fields</td>
</tr>
<tr>
<td>Name of a prompt in the subscribed report/document</td>
<td>{&amp;PromptNumber}</td>
<td>All fields</td>
</tr>
</tbody>
</table>

(where *Number* is the number of the prompt)
Managing your 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.

To manage your subscriptions in the Subscription Manager

1. In Desktop, 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 displays on the right-hand 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 Desktop, page 557.
   - 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 of the options 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 information about any of the options in this 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 456.

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 8, Clustering Multiple MicroStrategy Servers.

All subscriptions in a clustered system are load-balanced across all nodes that host the project containing the subscribed report.

Scheduling administrative tasks

In addition to scheduling the execution of reports and documents, you can instruct Intelligence Server to automatically 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, as well as any privileges required for that task.
To schedule an administrative task

1. In Desktop, 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 specific 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.

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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cache / History List management tasks</strong></td>
<td></td>
</tr>
<tr>
<td>Delete report caches</td>
<td>Delete all report caches for the project. For more information, see Deleting result caches, page 474. Note: Typically the “Invalidate Caches” task is sufficient to clear the report caches.</td>
</tr>
<tr>
<td>Delete History List messages (project or project source)</td>
<td>Delete all history list messages for the project or project source. For more information, see Scheduling History List message deletion, page 501. Note: This maintenance request can sometimes be large. Schedule History List deletions during times when Intelligence Server is not busy, such as during hours 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.</td>
</tr>
<tr>
<td>Invalidate caches</td>
<td>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 472.</td>
</tr>
<tr>
<td>Purge element caches</td>
<td>Delete the element caches for a project. For more information, see Deleting all element caches, page 516.</td>
</tr>
</tbody>
</table>
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 28.
To manage scheduled administration tasks

1. In Desktop, 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 8, Clustering Multiple MicroStrategy Servers*.

To determine which server handles each scheduled administrative task, use the following guidelines:

- Tasks 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 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 445*. 
Scheduling deliveries to email, file, and printer using Distribution Services

MicroStrategy Distribution Services is a software product that provides high-volume and high-efficiency distribution of reports, documents, and dashboards to email addresses, file servers, and networked printers. Distribution Services also supports various MicroStrategy Mobile-related features.

This chapter contains an explanation of 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 Desktop. For example, for steps that a 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 MicroStrategy Web’s Help topic “Creating alerts for specific threshold conditions”.

Distribution Services functionality is set up and enabled by an administrator in Desktop, 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.

For details about statistics logging for email, file, and print deliveries, see Statistics on subscriptions and deliveries, page 793.

About Distribution Services

Business users can set up information flows for themselves and other users by subscribing to report and document deliveries. Users view and change subscriptions created by or for them in their My Subscriptions page in MicroStrategy Web. Users can freely personalize 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, 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
Users can receive reports on a schedule, on an event, or on an alert basis, by designating various distribution triggers so that they only need to look at reports when the situation requires it. Triggers include:

- Send Now
- Time-based triggers, for example, scheduling distribution for every Monday morning or at the end of the fiscal quarter
- Event-based triggers, for example, scheduling distribution for when the books are closed or when data loading is complete
- Alert-based (exception threshold) triggers, for example, scheduling distribution for when inventory drops below 10% or when metrics indicate projected customer churn

Reports/documents that are subscribed to for delivery can be compressed and password-protected. The standard MicroStrategy security credentials are applied for each user subscribed to receive a report or document.

From an operations perspective, Distribution Services’ scheduling feature allows you to schedule reports and documents to run overnight during off-peak hours, thereby freeing up database and server resources for interactive use during peak hours of the day.

Distribution Services is integrated with Intelligence Server’s performance and stability features including:

- Clustering, job load balancing, execution control, fail-over, object caching, Intelligent Cubes, database optimization, multisource data access, Web-based user and contact administration
- Internationalization via reporting and delivery using multiple language data sources and metadata
- Administrative tools such as Command Manager for script-based administration, Enterprise Manager for enterprise BI analysis, and Object Manager for life cycle management.

**Administering report delivery**

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, or printers. These administrative tools include transmitters, devices, and contacts.
About transmitters

Transmitters are software components that Distribution Services uses to transform MicroStrategy reports and documents into emails and files and send those emails and files to email recipients, network file locations, or network printers. Distribution Services comes with default email, file, and print transmitters that you can use to create email, file, and print devices.

For details on how to create or modify transmitters, see Creating and managing transmitters, page 571.

About devices

Devices are 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 (based on the File transmitter) that specifies a particular 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, and print devices that are already set up, out of the box. 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 577.

About contacts

Contacts provide a user with a set of associated email addresses, file delivery location, and network printer delivery locations. 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. Contacts allow you to group multiple addresses together 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. Since a contact can be linked to only one MicroStrategy user account, no other users can access or see the address in a 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. For example, company shareholders may need to receive a financial document every month. One contact is created for each recipient’s email address, and then a MicroStrategy user is created, perhaps named Shareholder Finances, and 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 585.

About the delivery process

The following high-level steps show you how the subscription and delivery process work:

1. In Desktop, the administrator creates an email transmitter, a file transmitter, and/or a print transmitter. 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.

2. In Desktop, the administrator creates a device for each of the transmitters created above. All devices can have specific settings provided to support your specific 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.

3. In Desktop, the administrator creates contacts using the transmitters and devices created above. The contacts contain delivery addresses (email addresses, file paths, or printer paths), and are linked to a MicroStrategy user account for security purposes.
4 From MicroStrategy Web, a user identifies a report or document that they would like to subscribe to have delivered to them on a regular basis (either to an email address, to a network storage location, or to a printer.)

5 In MicroStrategy Web, the user selects his own address from the To drop-down menu. If he chooses, he can also select additional addresses for himself, other MicroStrategy users, or other contacts, to also receive this report or document based on 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.

Best practices for using Distribution Services features

- For best results, follow the steps listed in High-level checklist to set up a report delivery system, page 570.

- PDF, plain text, and CSV file formats generally experience the fastest delivery performance. Performance can vary, depending on a number of circumstances 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 reduced in size.
  - 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 during times when your Intelligence Server is experiencing low traffic.

- Establish reasonable limits on the number of scheduled jobs allowed. For details on this setting, see Limiting the total number of jobs, page 380.
• 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. See the Basic Reporting Guide for an introduction to creating and adding prompts to a report.

If your organization is processing a larger number of subscriptions, such as 1000 or higher, 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 a large number of subscriptions, you should consider using the bulk export feature. Details on bulk exporting are in the Reports chapter of the Advanced Reporting Guide.

• For best performance, consider configuring the following settings to suit your subscription needs:
  
  ‡ Tune the **Number of scheduled jobs** governing setting, based on 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. See the appropriate section of this manual for details on each setting.

• 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.

• When creating contacts, make sure that each contact has at least one address for each delivery type (email, file, and print). Otherwise the contact will 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, then when an email subscription is being created, that contact will 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 will be unable to run the report successfully and the subscription will be 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 580).

### High-level checklist to set up a report delivery system

#### 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.

#### Checklist

The following high-level checklist shows you what you need to do to set up a report delivery system in MicroStrategy using Distribution Services.

1. If you currently 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 Desktop **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*.

2. Modify existing transmitters or create new transmitters according to your requirements. Distribution Services comes with default email, file, and print transmitters that are already set up, 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 572.

• For steps to modify a transmitter, see Viewing and modifying a transmitter and accessing the Transmitter Editor, page 573.

• For steps to create a new transmitter, see Creating a transmitter, page 573.

3 Modify existing devices or create new devices according to your requirements. Distribution Services comes with default email, file, and print devices that are already set up, but if you use these you should modify their settings to suit your network, email system, and printer systems in your environment.

• For best practices for working with devices, see Best practices for working with devices, page 578.

• For steps to modify a device, see Viewing and modifying a device and accessing the device editors, page 579.

• For steps to create a new device, see Creating and managing devices, page 577.

4 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 587.

• For steps to modify a contact, see Viewing and modifying a contact and accessing the Contact List area, page 587.

• For steps to create a contact, see Creating a contact, page 589.

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 three types of transmitters: email (SMTP), file, and print. 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, while others might specify UUEncoding. For information on devices and their settings, see Creating and managing devices, page 577.

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 (email, file, and print) 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 Desktop 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. Click Help for details on each option in the interface.
5. Click OK to save any changes.

Creating a transmitter

In MicroStrategy Desktop, 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.

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 whenever you need a specific combination of properties and settings for a file, email, or print 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 specific needs for this new transmitter. This is a time-saving method if you have a similar transmitter already created, or you want to make use of the default MicroStrategy transmitter. To duplicate a transmitter, right-click the transmitter that you want to duplicate and select **Duplicate**. Click **Help** for details on each option in the interface.

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 recipient(s) who subscribed to the file.

**To create an email transmitter**

1. From the Desktop 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.
4. Change the transmitter settings as desired. Click Help for details on each option in the interface.
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 an email transmitter is created, you can create email devices 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 email transmitter apply to all email devices that will be based on this transmitter.
Creating a file transmitter

A file transmitter transforms a subscribed report or document into a file format which is chosen by the user 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 Desktop 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. Click Help for details on each option in the interface.

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 file transmitter is created, you can create file devices 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 this transmitter.

For information on creating a file device, see Creating a file device, page 579.

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 Desktop 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. Click Help for details on each option in the interface.

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 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 print transmitter apply to all print devices that will be based on this transmitter.

For information on creating a print device, see Creating a print device, page 583.

Deleting a transmitter

You can delete a transmitter if you no longer need to use it.

Prerequisites

- You cannot delete a transmitter if there are devices depending on the transmitter. You must first delete any devices that depend on the transmitter.

To delete a transmitter

1. From the Desktop 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, then 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 MicroStrategy Desktop, 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.

- **File**: A file device automatically sends a MicroStrategy report or document in a file format chosen by users while 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 579*.

- **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 583*.
You create new devices whenever you need a specific combination of properties and settings for a file, email, or print device to deliver files. There are two ways to create a new device. 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.
- Test a delivery using each device (email, file, and print) to make sure that the device settings are still effective and any system changes that may 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 Desktop 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. Click **Help** for details on each option in the interface.

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.

**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, which 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 whenever 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 specific 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 file device. To duplicate a device, right-click the device that you want to duplicate and select **Duplicate**. Click **Help** for details on each option in the interface.

---

**To create a new file device**

1. From the Desktop 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. Click **Help** for details on each option in the interface.

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. Click **Help** for details to create an address for a user or to create a contact and add addresses to the contact.

---

**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}\` then all subscriptions using that file device are delivered to subfolders of `C:\Reports\`. Subscribed reports/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.

The table below lists the macros that can be used in the **File Location** and **Backup File Location** fields in a file device:

<table>
<thead>
<tr>
<th>Text</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date on which the subscription is sent</td>
<td>{&amp;Date}</td>
</tr>
<tr>
<td>Time at which the subscription is sent</td>
<td>{&amp;Time}</td>
</tr>
<tr>
<td>Name of the recipient</td>
<td>{&amp;RecipientName}</td>
</tr>
<tr>
<td>User ID (32-character GUID) of the recipient</td>
<td>{&amp;RecipientID}</td>
</tr>
<tr>
<td>Distribution Services address that the subscription is delivered to</td>
<td>{&amp;AddressName}</td>
</tr>
</tbody>
</table>

**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 the 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 purchasing, installing, and using 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.

2. Create a new file device or edit an existing file device (see *Creating a file device, page 579*).
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 will require a device with different settings than 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 specific 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. Click Help for details on each option in the interface.

Prerequisites

• An understanding of your organization’s email server or other email delivery systems.
To create a new email device

1. From the Desktop 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.

4. Change the device settings as desired. Click Help for details on each option in the interface.

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. Click Help for details to create an address for a user or to create a contact and add addresses to the contact.

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 out. 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.

Prerequisites

- The selected printer must be added to the list of printers on the machine on which MicroStrategy Intelligence Server is running.
To create a new print device

1. From the Desktop 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. Click Help for details on each option in the interface.

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 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. Click Help for details to create an address for a user or to create a contact and add addresses to the contact.

Deleting a device

You can delete a device if you no longer need to use it.

Prerequisites

Update those 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, use the Delivery Manager for 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 Desktop 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 Objects message is displayed. See the Prerequisites above to be sure you have properly prepared the system to allow the device to be deleted.

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, or a network printer path). 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, or printer 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 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 Desktop Help.

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 within that contact.

Contacts can also be combined into groups, so that it is easier to select multiple contacts for a given subscription.
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, which require Flash to be installed wherever the dashboard is delivered to. For specific 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

Recommended maintenance tasks for contacts

- Verify email addresses, file storage locations, and printer locations 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 delivered to. 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 might 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 based on users who will receive the same subscribed reports. Tree View can be helpful as your list of contacts grows larger, since 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. Click **Help** for steps to use the filter.

• 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 impacted by your action.

**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 Desktop 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 the contact:

- **Duplicate**: Creates a copy of the contact. See Creating a contact, page 589.

- **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 592.

- **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 period of 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.
Creating a contact

You create a new contact for each address (email address, file storage location on your network, or network printer path) 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 specific needs for this new contact. This is a time-saving method if you have a similar contact already created, 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. Click Help for details on each option in the interface.

To create a contact

1. From the Desktop 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. Click Help for details on each option in the interface.

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. Since 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, double-click a user to open the account in the User Editor, and use the Addresses tab to link the user to a contact. Click Help for details on each option in the interface.

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 there are certain reports that need to be sent to multiple contacts. For example, if there are four contacts that 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. Since 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 Desktop 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. Click **Help** for details on each option in the interface.

   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.

All members (contacts) of each contact group within 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. Since 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 592.

- **Disable Contact**: Disables user’s delivery locations (addresses). If a user’s addresses are disabled, the user cannot be subscribed to reports. If there are reports 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.

- **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 for file storage, a printer path, or an email address.

A MicroStrategy user can have several email, file, and/or printer 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 to a user on the **Addresses** tab of the User Editor.

In the Contacts List area, right-click and address for a contact or a user and select from the following options:

- **Edit**: Opens the User Editor: Addresses tab if you right-click an address within a user. Opens the Contact Editor: Addresses tab if you right-click an address within a contact. Click **Help** for details on each option in the interface.

- **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 Desktop 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.

Subscription and delivery tasks for users

The table below lists common subscription and delivery-related tasks that users or administrators can perform, and where they perform those tasks, in both MicroStrategy Web and Desktop. Note that some tasks can only be performed 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.

<table>
<thead>
<tr>
<th>User Task</th>
<th>Where to Perform the Task</th>
</tr>
</thead>
</table>
| Define preferences for a report/document to be delivered to user’s History List folder, mobile device, or system cache. | • In Desktop: In a report or document, from the File menu, select Schedule delivery to, then select History List, Update cache, or Mobile.  
• In Web: In a report or document, from the Home menu, select Subscribe to, then select History List or Mobile. |
<p>| Define preferences for a report/document to be delivered to an email address, network storage location, or printer. | • In Web, from a report or document, from the Home menu, select Subscribe to, then select Email, File, or Printer. |</p>
<table>
<thead>
<tr>
<th>User Task</th>
<th>Where to Perform the Task</th>
</tr>
</thead>
</table>
| Define personal subscription preferences to all reports or documents, in one location. | • In Desktop: From the **Tools** menu, select **My Subscriptions**.  
• In Web: Click **Preferences** at the top of any page. For History List delivery, select **Project Defaults** on the left, then select **History List**. For Email, File, or Printer delivery, select **User Defaults** on the left, then select **Email addresses**, **File locations**, or **Printer locations**. |
| Set up alert-based subscriptions.                                        | In 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 user’s History List or mobile device. For Web, schedule a delivery to an email address, a network file storage location, or a printer. | In Desktop: In a report or document, from the **File** menu, select **Schedule delivery to**, then select **History List**, **Update cache**, or **Mobile**.  
In Web:  
• If you do not own Distribution Services: In a report or document, from the **Home** menu, select **Add to History List** or **Add to Mobile**.  
• If you own Distribution Services: In a report or document, from the **Home** menu select **Subscribe to**, then select **History List**, **Mobile**, **Email**, **File**, or **Printer**.                                                                                          |
| Unsubscribe from a report or document.                                   | • In Desktop: From the **Tools** menu, select **My Subscriptions**. Right-click a subscription, then select **Unsubscribe**.  
• In Web: click **My Subscriptions** at the top of any page. In the Unsubscribe column on the right, select a check box and click **Unsubscribe**.                                                                                                    |
| Change subscription details for a report or document.                   | • In Desktop: From the **Tools** menu, select **My Subscriptions**. Right-click a subscription, then select **Edit**.  
• In Web, click **My Subscriptions** at the top of any page. In the Action column, click the **Edit** icon for the report/document whose subscription you want to edit.                                                                                                           |
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 within 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 MicroStrategy Desktop or Narrowcast Administrator interface. In addition, you can 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

For example, you can change the system to allow more low 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 602.

Other examples of tasks you can perform using Command Manager include:

- **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.

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 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 598), and the presence of script outlines (see Script outlines, page 598).

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 more information about the Command Manager command line interface, including instructions, see Using Command Manager from the command line, page 609.

To start the Command Manager graphical interface

In Windows: From the Windows Start menu, point to Programs, then MicroStrategy, then Command Manager, 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, from the Help menu in the graphical Command Manager, select Command Manager Help.
Color-coding the text in a script

The Command Manager script editor has the ability to color-code the text according to its function in the script or procedure.

In a Command Manager script:
• Reserved words appear in blue.
• Words or phrases in quotation marks appear in gray.
• Numbers appear in red. Dates appear in red with blue slashes.
• GUIDs appear in green.
• All other text appears in black.

In a Command Manager procedure:
• Keywords, such as if or boolean, appear in purple, and bold.
• Functions, classes, and methods appear in red.
• Command Manager statements appear in blue.
• Comments appear in 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 pre-constructed statements with optional features and user-defined parameters clearly marked.

Outlines are grouped based on 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 outlines appropriate 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 outline as needed for your script.

**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");
```

where NewUser is the name of the procedure, and KHuang and Customers are the inputs to the procedure.

Procedures are only available for use with project sources. Procedures cannot be used with Narrowcast Server statements.

Command Manager contains a large number of sample procedures that you can view and modify. These are stored in the Command Manager directory, in `\Outlines\Procedure_Outlines\Sample_Procedures\`

For detailed 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 only supported in procedures, and procedures are only supported 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 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.

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 597)
From the Command Manager command line interface (see Using Command Manager from the command line, page 609)

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 609.

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 604.

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 Programs, then MicroStrategy, then Command Manager, 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.
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.

<table>
<thead>
<tr>
<th>Effect</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection (required; choose one)</strong></td>
<td></td>
</tr>
<tr>
<td>Connect to a project source</td>
<td><code>-n ProjectSourceName</code>&lt;br&gt;<code>-u UserName</code>&lt;br&gt;<code>[-p Password]</code></td>
</tr>
<tr>
<td><strong>Note:</strong> If <code>-p</code> is omitted, Command Manager assumes a null password.</td>
<td></td>
</tr>
<tr>
<td>Initiate a connection-less project source session</td>
<td><code>-connlessMSTR</code></td>
</tr>
<tr>
<td>Connect to a Narrowcast Server</td>
<td><code>-w ODBC_DSN</code>&lt;br&gt;<code>-u UserName</code>&lt;br&gt;<code>[-p Password]</code>&lt;br&gt;<code>-d Database</code>&lt;br&gt;<code>[-s SystemPrefix]</code></td>
</tr>
<tr>
<td><strong>Note:</strong> If <code>-p</code> or <code>-s</code> are omitted, Command Manager assumes a null password or system prefix.</td>
<td></td>
</tr>
<tr>
<td>Initiate a connection-less Narrowcast Server session</td>
<td><code>-connlessNCS</code>&lt;br&gt;<code>[-d Database]</code>&lt;br&gt;<code>[-s SystemPrefix]</code></td>
</tr>
<tr>
<td><strong>Note:</strong> If <code>-s</code> is omitted, Command Manager assumes a null system prefix.</td>
<td></td>
</tr>
</tbody>
</table>
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:

Program Files\MicroStrategy\Command Manager

<table>
<thead>
<tr>
<th>Effect</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Script input (required)</strong></td>
<td></td>
</tr>
<tr>
<td>Specify the script file to be executed</td>
<td>-f InputFile</td>
</tr>
<tr>
<td><strong>Note:</strong> If this parameter is omitted, the Command Manager GUI is launched.</td>
<td></td>
</tr>
<tr>
<td><strong>Script output (optional; choose only one)</strong></td>
<td></td>
</tr>
<tr>
<td>Log script results, error messages, and status messages to a single file</td>
<td>-o OutputFile</td>
</tr>
<tr>
<td>Log script results, error messages, and status messages to separate files, with default file names of:</td>
<td>-break</td>
</tr>
<tr>
<td>• CmdMgrResults.log</td>
<td></td>
</tr>
<tr>
<td>• CmdMgrFail.log</td>
<td></td>
</tr>
<tr>
<td>• CmdMgrSuccess.log</td>
<td></td>
</tr>
<tr>
<td>Log script results, error messages, and status messages to separate files, with specified names</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> You can omit one or more of these parameters. For example, if you only want to log error messages, use only the -of parameter.</td>
<td></td>
</tr>
<tr>
<td><strong>Script output options (optional)</strong></td>
<td></td>
</tr>
<tr>
<td>Begin each log file with a header containing information such as the version of Command Manager used</td>
<td>-h</td>
</tr>
<tr>
<td>Print instructions in each log file and on the console</td>
<td>-i</td>
</tr>
<tr>
<td><strong>Note:</strong> This option is ignored if the script is encrypted. For information about encrypted Command Manager scripts, see Encrypting Command Manager scripts, page 604.</td>
<td></td>
</tr>
<tr>
<td>In case of an Intelligence Server error, print the Intelligence Server error code in each log file and on the console</td>
<td>-e</td>
</tr>
<tr>
<td>Display script output on the console</td>
<td>-showoutput</td>
</tr>
<tr>
<td>Save the results of script instructions in an XML file</td>
<td>-xml XMLFile</td>
</tr>
<tr>
<td>Omit hidden objects in the script results. Hidden objects are MicroStrategy metadata objects whose HIDDEN property is set.</td>
<td>-suppresshidden</td>
</tr>
<tr>
<td><strong>Execution options (optional)</strong></td>
<td></td>
</tr>
<tr>
<td>Halt script execution on critical errors (see Handling execution errors, page 606)</td>
<td>-stoponerror</td>
</tr>
</tbody>
</table>
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 username 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. Since 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. In addition, it 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, there is the possibility 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 MicroStrategy Desktop 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, they see a message that informs them that the project or configuration is locked because another user is modifying it. Users can then choose to open the object in read-only mode or view more details about the lock.

There are two kinds of locks in Command Manager:

- **Transient** locks are automatically released after disconnecting.
- **Permanent** locks are only released 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 Desktop Help (from within Desktop, press **F1**).

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 is not able to create the user. Since 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.

- **Non-critical errors** occur when the main part of the instruction is able to complete. These errors never interrupt script execution.

  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 is still able to create the group. Since creating the group is the main part of the instruction (adding users is secondary), this is a non-critical error.

An error message is written to the Messages tab of the Script window for all execution errors, critical or non-critical. 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 is not able to manipulate prompted objects. For example, it cannot alter the properties of a metric that contains a prompt, and 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 non-critical 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

**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

- **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.

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.
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 information about performance tests, see *Testing Intelligence Server performance, page 614.*

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 612*
- *Best practices for using Integrity Manager, page 616*
- *Creating an integrity test, page 617*
- *Executing an integrity test, page 620*
- *Viewing the results of a test, page 631*

### 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 619.*

**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.

**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.

- 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 Generation 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 628*.

- 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 621*.
  - 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 625*. 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 differences in graph and PDF reports, see Grouping differences in graph and PDF reports, page 633.

- If you are executing a performance test, follow the best practices described in Performance test best practices, page 615.

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 Start menu, point to Programs, then MicroStrategy, then Integrity Manager, 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 integrity test.
   - To compare reports and documents against a previously established baseline, select Baseline versus project integrity test.
   - To compare reports and documents from two previously established baselines, select Baseline versus baseline integrity test.
   - To confirm that reports and documents in a project execute without errors, select Single project integrity test.
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 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 619.

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 617*.

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 628*.

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**.

### 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. 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 Prompt 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 Prompt 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 Prompt 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 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:

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prompt1</td>
<td>AnswerA, AnswerB</td>
</tr>
<tr>
<td>Prompt2</td>
<td>AnswerA, AnswerC, AnswerD</td>
</tr>
</tbody>
</table>

Integrity Manager executes this report four times, as shown in the table below:

<table>
<thead>
<tr>
<th>Execution</th>
<th>Prompt 1 answer</th>
<th>Prompt 2 answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Personal answer AnswerA</td>
<td>Personal answer AnswerA</td>
</tr>
<tr>
<td>2</td>
<td>Personal answer AnswerB</td>
<td>(next prompt answer method)</td>
</tr>
<tr>
<td>3</td>
<td>(next prompt answer method)</td>
<td>Personal answer AnswerC</td>
</tr>
<tr>
<td>4</td>
<td>(next prompt answer method)</td>
<td>Personal answer AnswerD</td>
</tr>
</tbody>
</table>

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
- 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 Prompt 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 Prompt 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 Prompt 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 text is executed, the reports are executed in the following order:

<table>
<thead>
<tr>
<th>Report execution</th>
<th>Base project report and user</th>
<th>Target project report and user</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Report1 Alice</td>
<td>Report1 Bob</td>
</tr>
<tr>
<td>2</td>
<td>Report2 Alice</td>
<td>Report2 Bob</td>
</tr>
<tr>
<td>3</td>
<td>Report3 Alice</td>
<td>Report3 Bob</td>
</tr>
<tr>
<td>4</td>
<td>Report1 Carol</td>
<td>Report1 Alice</td>
</tr>
<tr>
<td>5</td>
<td>Report2 Carol</td>
<td>Report2 Alice</td>
</tr>
<tr>
<td>6</td>
<td>Report3 Carol</td>
<td>Report3 Alice</td>
</tr>
</tbody>
</table>

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. For more information about dynamic SQL, consult your database reference.

You can force Integrity Manager to ignore the dynamic SQL in its comparison by using the Ignore Dynamic SQL feature. Access this feature from the Advanced Options dialog box, on the Select Processing Options page of the Integrity Manager wizard.

If you have multiple sections of dynamic SQL, you can force Integrity Manager to ignore all of them by inserting identifying SQL comments before and after each section of dynamic SQL. For example, before each section of dynamic SQL you could include the line

```sql
/* BEGIN DYNAMIC SQL */
```

and after each section you could include the line

```sql
/* END DYNAMIC SQL */
```

Then you can enter `/* BEGIN DYNAMIC SQL */` in the **Start** field, and `/* END DYNAMIC SQL */` in the **End** field.
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.

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 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 619), call the Integrity Manager executable MIntMgr.exe. By default this executable is located in C:\Program Files\MicroStrategy\Integrity Manager.

The syntax is:

MIntMgr.exe -f FileLocation\Filename.mtc [-bp 'BasePassword1', 'BasePassword2', ..., 'BasePasswordn'] [-tp 'TargetPassword1', 'TargetPassword2', ..., 'TargetPasswordn']

where:

- **FileLocation** is the path to the saved test file.
- **Filename.mtc** is the name of the saved test file.
- **BasePassword1..n** are the passwords for the users specified in the test file to log in to the base project, in the order that they are specified in the test file, in single quotes and separated by commas. This is not required for a baseline-versus-project or baseline-versus-baseline integrity test. For more information about executing a test with multiple users, see Executing a test under multiple MicroStrategy user accounts, page 625.
- **TargetPassword1..n** are the passwords for the users specified in the test file to log in to the target project, in the order that they are specified in the test file, in single quotes and separated by commas. This is not required for a single-project or baseline-versus-baseline integrity test.

Note the following:

- Each password must be enclosed in single quotes (' ') and the passwords must be separated by a comma. If a password contains a single quote or a comma, that character must be preceded by a single quote. For example, the password asdf'asdf,asdf must be typed as 'asdf''asdf',asdf'.
- If a user in the base project or target project has an empty password, the position in the list of passwords where that user's password would be 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'

For example, to run a saved single-project test file named MonthlyTest.mtc located in C:\Tests for which the passwords for the specified users in the base project are admin123 and admin5, use the following command:

MIntMgr.exe -f C:\Tests\MonthlyTest.mtc -bp 'admin123', 'admin5'

For syntax information for the Windows AT command or a UNIX scheduler, see the documentation for your operating system.

**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 text execution, type `echo %ERRORLEVEL%` and press Enter.

<table>
<thead>
<tr>
<th>Exit code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The test execution succeeded and all reports have a status of Matched.</td>
</tr>
<tr>
<td>1</td>
<td>The test execution succeeded, but at least one report has a status other than Matched.</td>
</tr>
<tr>
<td>2</td>
<td>Integrity Manager was unable to establish a connection to Intelligence Server, or the connection was interrupted during the test.</td>
</tr>
<tr>
<td>3</td>
<td>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.</td>
</tr>
<tr>
<td>4</td>
<td>The test execution failed. For more information about this error, see the integrity test log for this test.</td>
</tr>
<tr>
<td>5</td>
<td>The test execution was aborted by the user.</td>
</tr>
</tbody>
</table>
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 cancelled 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 621*. 
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 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 reports.

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 Desktop.
- 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.

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:

**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 **Put output in a subfolder named by execution date/time** 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 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
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.
ANALYZING SYSTEM USAGE WITH ENTERPRISE MANAGER

Introduction

MicroStrategy Enterprise Manager provides insights about governing and tuning all areas of your MicroStrategy environment. By analyzing the usage statistics collected by Intelligence Server, Enterprise Manager provides a historical overview of Intelligence Server operations.

This section provides the following information:

• What is Enterprise Manager?, page 640 provides an overview of Enterprise Manager.

• Best practices for using Enterprise Manager, page 641 provides MicroStrategy’s recommendations for best practices when using Enterprise Manager.

• Understanding the Enterprise Manager architecture, page 641 explains how Enterprise Manager interacts with your MicroStrategy system.

• Installing and configuring Enterprise Manager, page 645 walks you through the process of setting up Enterprise Manager to monitor projects in your system.
• *Maintaining Enterprise Manager, page 659* explains how Enterprise Manager uses the data in the Intelligence Server statistics tables.

• *Reporting in Enterprise Manager, page 669* provides descriptions of all the predefined reports available in Enterprise Manager. These reports can help you identify problem areas in the system.

## What is Enterprise Manager?

The heart of Enterprise Manager is a MicroStrategy project that comes with a set of predefined reports, and the metrics and attributes to create your own reports. When you log into the Enterprise Manager project, you can run reports that help you to:

• Allocate system resources based on data warehouse usage

• Research efficient aggregation, partitioning, and indexing strategies

• Determine the optimal time to run scheduled jobs, load data, or perform system and database maintenance

• Determine the most popular reports so you can schedule and cache them, thus increasing their response time and reducing the load on the system

• Identify unused objects from your metadata repository so they can be deleted later

• Identify peak usage times and patterns and, if necessary, tune your Intelligence Server to respond appropriately

• Determine whether you need to add more threads to the database connection threads if queue times are long

• Profile users based on their system resource usage

You can use the predefined reports as they are, copy the reports and then modify the copies, or build your own custom reports to suit your needs. The Enterprise Manager project includes over 300 metrics and 90 predefined reports. Many of these reports include prompts, which accept user input before a report is executed, for flexibility. You can create new metrics, prompts, filters, templates, or reports to suit the needs of your environment or the type of analysis you want to do. All of the predefined objects are located in the Public Objects folder in the Enterprise Manager project.
Best practices for using Enterprise Manager

- Make all users who need access to the Enterprise Manager reports members of the MicroStrategy Web Viewer user group. Users in this group have all the necessary permissions and privileges to use the out-of-the-box Enterprise Manager reports.

- Make all users who need administrative access to the Enterprise Manager project members of the EMAdmin user group. Users in this group have all the necessary permissions and privileges to administer the Enterprise Manager project.

- Use Enterprise Manager to monitor itself. This feedback can help you fine-tune Enterprise Manager’s monitoring ability. For instructions on how to monitor a project in Enterprise Manager, see *Selecting the projects to monitor, page 655*.

- For additional information about every object in the Enterprise Manager project, see the object’s Long Description property (right-click the object, select *Properties*, and select the *Long Description* category). The long description includes sample reporting requirements for the object, where applicable.

- For best practices relating to specific Enterprise Manager functionality, see the following sections:
  - *Best practices for installing and configuring Enterprise Manager, page 646*
  - *Best practices for Enterprise Manager data loading, page 661*
  - *Best practices for Enterprise Manager reporting, page 671*

Understanding the Enterprise Manager architecture

The diagram below shows the main components of Enterprise Manager, and the processes by which Intelligence Server statistics are incorporated into Enterprise Manager reports. Descriptions of these components and processes are below the diagram, organized into the following categories.

- *Enterprise Manager system components, page 642*
• **Enterprise Manager processes, page 643**

Enterprise Manager system components

- **Intelligence Servers**

  The projects on each Intelligence Server can be configured to log usage statistics to the statistics database. The logged statistics include information such as the report GUID, the user who ran the report, how long various stages of the report took to execute, and so on.

- **Enterprise Manager**

  Enterprise Manager consists of an application (the Enterprise Manager console) and a project.
The Enterprise Manager console loads information about the monitored projects into the Enterprise Manager data warehouse lookup tables, and runs the data load scripts that move statistics data into the data warehouse fact tables. For detailed information about the Enterprise Manager console, see Running the Enterprise Manager console, page 650, or see the Enterprise Manager console Help.

The Enterprise Manager monitoring project’s data warehouse is populated from the Intelligence Server statistics. The reports in this project are designed to assist users in tuning their entire MicroStrategy system. For detailed descriptions of each report, see Reporting in Enterprise Manager, page 669.

• **Statistics tables**

  The statistics database contains MicroStrategy-specific statistics tables created through the MicroStrategy Configuration Wizard. These tables contain data on the MicroStrategy system’s usage and performance, and are populated by all projects that are configured to log statistics. For an explanation of the collected statistics, see Monitoring system usage: Intelligence Server statistics, page 323. For detailed information about the contents of the statistics tables, see the Statistics Data Dictionary in the Reference for System Administration.

• **Enterprise Manager data warehouse**

  The Enterprise Manager data warehouse is in the same physical database as the statistics tables. It contains configuration tables necessary for Enterprise Manager’s operation, lookup tables, and fact tables with transformed information from the statistics tables.

  ‡ **Lookup tables** contain descriptive information about each object in the monitored projects, such as name, owner, creation date, folder path, and so on.

  ‡ **Fact tables** contain data that has been loaded from the statistics database by the data load process.

**Enterprise Manager processes**

The following subsections describe the three major Enterprise Manager processes: logging statistics, loading data, and reporting on that data.
The statistics logging process

You choose the MicroStrategy projects to log usage statistics into the statistics tables. For details about the statistics logging process, including instructions on configuring a project to log statistics, see the Monitoring the System chapter in the System Administration Guide.

The data loading process

Before the raw information in the statistics database can be analyzed with the monitoring project, it must be converted. In addition, Enterprise Manager needs up-to-date information about the projects it monitors to report accurately on topics like per-user resource usage. The data load process populates both the lookup and fact tables in the Enterprise Manager data warehouse. For a detailed description of the data load process, see Data loading, page 660

The data load process is broken into two steps.

1  In the first step, Enterprise Manager retrieves information about the structure of the monitored projects by means of the Intelligence Server API. It then populates the lookup tables of the data warehouse with that information.

2  In the second step, Enterprise Manager migrates data from the statistics tables into the data warehouse fact tables by means of a set of SQL scripts.

The Enterprise Manager reporting process

The Enterprise Manager administrator executes reports in the Enterprise Manager project to analyze the information in the data warehouse. For detailed descriptions of each report, see Reporting in Enterprise Manager, page 669.
Installing and configuring Enterprise Manager

Below is a high-level overview of the steps to install and configure Enterprise Manager for your MicroStrategy system.

1 Confirm that you have fulfilled the prerequisites for installing Enterprise Manager.

   For a list of Enterprise Manager’s prerequisites, see MicroStrategy Enterprise Manager prerequisites, page 647.

2 Install the Enterprise Manager console and databases.

   For more information about installing Enterprise Manager, see Installing Enterprise Manager, page 648. For detailed instructions on installing MicroStrategy products, see the MicroStrategy Installation and Configuration Guide.

3 Configure your projects to log statistics to the statistics tables.

   For each project, you need to set the location of the statistics tables and the information that you want to have logged. For more information and detailed instructions, see the Monitoring the System chapter in the System Administration Guide.

   The Enterprise Manager data warehouse is stored in the same database that is used to store the statistics.

4 Set up the Enterprise Manager metadata database.

   This database contains the users, groups, objects, and reports for the Enterprise Manager project. For more information and detailed instructions, see Creating the Enterprise Manager metadata database, page 649.

5 Step through the Enterprise Manager Console, which sets up the following processes:

   a Initialize, which migrates the Enterprise Manager project to your Enterprise Manager production databases. For more information and
detailed instructions, see *Initializing your Enterprise Manager production environment, page 650.*

b Configure, which selects the projects to monitor. For more information and detailed instructions, see *Selecting the projects to monitor, page 655.*

c Schedule, which specifies how often your projects’ statistics are loaded into the Enterprise Manager data warehouse. For more information and detailed instructions, see *Scheduling how often statistics are loaded into the Enterprise Manager data warehouse, page 657.*

**Best practices for installing and configuring Enterprise Manager**

- Install Enterprise Manager on a machine that is separate from Intelligence Server. This configuration allows you to upgrade Enterprise Manager without upgrading Intelligence Server. You can have the Enterprise Manager project in the same metadata as your other projects.

- The Enterprise Manager project metadata contains two user groups: MicroStrategy Web Viewer and EMAdmin.
  - Users assigned to the MicroStrategy Web Viewer group have all the necessary privileges and permissions to view the out-of-the-box Enterprise Manager reports. Assign users who only need to view reports to this group.
  - Users assigned to the EMAdmin group have all the necessary privileges to administer the Enterprise Manager project. Assign users who need to create reports to this group.

- To ensure that you can successfully upgrade the Enterprise Manager project in the future, do not modify schema objects. Rather, make copies of the objects you want to modify and then modify the copies.

- Upgrade to Enterprise Manager service packs when they become available. MicroStrategy includes your feedback in the service packs, including fixes to issues and additional enhancements.
MicroStrategy Enterprise Manager prerequisites

Before you install Enterprise Manager, make sure you have the following:

- Write access to two databases: one for the data warehouse and statistics tables, and one for the Enterprise Manager project metadata. The project metadata can be placed in the same database as other MicroStrategy project metadatas.

The Enterprise Manager data warehouse must be placed in the same database that is used to store Intelligence Server statistics. It can be stored in any of the following databases:

- SQL Server
- Oracle
- Teradata
- IBM DB2 UDB
- Sybase ASE

If you are using DB2 or Sybase for the data warehouse, you must change the minimum page file size for the database to at least 8K. DB2 uses the page file size to determine the maximum size of a single row in the database. Some Enterprise Manager tables contain rows that require more than the default 4K.

For a complete list of the versions of each database that are certified for use with Enterprise Manager, see the Enterprise manager readme file.

- Administrator privileges for the MicroStrategy projects that you want to monitor in Enterprise Manager. You must also have the Create Configuration Objects privilege for the project source in which you are creating the Enterprise Manager project.

- The frequency and best times for Enterprise Manager to load data from the statistics tables into the data warehouse. For example, if you need near-real-time data, you can load the data as often as once per hour. However, if you are collecting statistics in great detail, you should run the data load during times when Intelligence Server usage is low, such as overnight. For an explanation of the data loading process, see Data loading, page 660.

- To use the dashboards that come with Enterprise Manager, you need a MicroStrategy Report Services license. If you do not have a Report Services license, contact your account representative for information
about obtaining one. For information about the dashboards in Enterprise Manager, see *Dashboards, page 673*.

These dashboards are designed for use with MicroStrategy Web. If you are using MicroStrategy Web Universal, the links to other reports in the dashboards do not function. To correct the links, edit the dashboards and change all occurrences of `Main.aspx` in the links to `mstrWeb`.

**Installing Enterprise Manager**

Enterprise Manager is one of the products provided with MicroStrategy. For information about obtaining a license to use Enterprise Manager, contact your MicroStrategy account executive.

The following Enterprise Manager components are installed:

- Microsoft Access™ metadata database: This database contains the metadata for the Enterprise Manager project, including all of the facts, attributes, hierarchies, metrics, filters, and reports that are predefined as part of Enterprise Manager.

- Microsoft Access data warehouse database: This database contains the fact and lookup tables that need to be populated, as well as configuration tables with information necessary for the operation of Enterprise Manager.

- Enterprise Manager Console: This application helps you configure Enterprise Manager, and launches the data loading scripts.

- SQL scripts: A wide variety of SQL scripts that create warehouse tables, perform various other setup tasks, and perform the data load process.

For instructions on installing Enterprise Manager, see the *MicroStrategy Installation and Configuration Guide*.

**Configuring your projects to log statistics**

Before Enterprise Manager can be configured to read statistics data, you must create the statistics database and, for each project, set the location of the statistics tables and the information that you want to have logged. For more information and detailed instructions, see *Configuring Intelligence Server to log statistics, page 330*. 
Creating the Enterprise Manager metadata database

The Enterprise Manager metadata database contains all the facts, attributes, hierarchies, metrics, filters, and reports that are predefined as part of the Enterprise Manager project. You can use an existing metadata database in your system, or create a new database:

- To use an existing metadata database, note the Data Source Name (DSN) for it. This DSN is used later in the installation process, when you are prompted for the Metadata Repository Location.

- To create a new database, follow the procedure below.

The Enterprise Manager-specific tables in this database are created and populated when you proceed through the Enterprise Manager Console, in the Initialization step. This is done by importing the Microsoft Access metadata database into the production database. For information about this process, see Initializing your Enterprise Manager production environment, page 650.

To create a new Enterprise Manager metadata database

1. Create a database to store your Enterprise Manager metadata. (This is generally performed by your database administrator.) This database must be one of the databases certified for use with MicroStrategy metadata. For a list of certified databases, see the MicroStrategy readme.

2. Use the MicroStrategy Connectivity Wizard to create a Data Source Name for the metadata. Note this DSN for later. It is needed when creating the statistics tables and when you specify the DSN for the Metadata Location.

To access the Connectivity Wizard, from the Windows Start menu point to Programs, then MicroStrategy, then Tools, then select Connectivity Wizard. For detailed instructions on using the Connectivity Wizard, see the MicroStrategy Installation and Configuration Guide.

Now you must create a data warehouse to store the statistics that Enterprise Manager will help you analyze.
Running the Enterprise Manager console

The Enterprise Manager console provides a wizard that walks you through the steps to initialize and configure the Enterprise Manager project. It consists of three major sections:

- *Initializing your Enterprise Manager production environment, page 650*
- *Selecting the projects to monitor, page 655*
- *Scheduling how often statistics are loaded into the Enterprise Manager data warehouse, page 657*

If you are using complete session logging, you must follow these steps for each different statistics database. For information about how to use the recommended configuration of single instance session logging, see the *Monitoring the System* chapter in the *System Administration Guide*.

You can close the Enterprise Manager Console at any time without losing anything you have already completed. The Enterprise Manager Console automatically saves all the work that you do and continues from where you stopped when you open the console again.

Initializing your Enterprise Manager production environment

The Enterprise Manager metadata and data warehouse that are installed with Enterprise Manager are in Microsoft Access databases. To use them in a production environment, you need to import them into the Enterprise Manager databases that you created earlier.

The metadata contains all of the Enterprise Manager project definition information: schema objects, metrics, templates, filters, reports, and more. The data warehouse contains configuration tables necessary to the operation of Enterprise Manager, some time-based lookup tables, and standard lookup tables that are populated during the data load process.

Initializing the environment also involves running a SQL script specific to the type of production database that you are using. This SQL script creates the lookup and fact tables and creates the primary keys in the data warehouse. It also inserts database-specific SQL procedures into the EM_SQL table for use in the data load process.
In the event of problems with the initialization process, you can check the file EMProjectMover.log to help with troubleshooting. This file is located by default in C:\Program Files\MicroStrategy\Enterprise Manager.

To initialize the Enterprise Manager project

1. From the Start menu, point to Programs, then MicroStrategy, then Enterprise Manager, then choose Enterprise Manager Console.

2. Read the Home page and click Next.

3. Read the Initialize Overview page. Be sure you have the Data Source Names for the Enterprise Manager data warehouse and the Enterprise Manager metadata database before you continue. Click Next.

4. Read the Initialize Project page and click Transfer to import the project files. The Project Mover Wizard opens.

   This procedure walks you through the Project Mover Wizard. For a detailed explanation of the wizard, see the Help for the Project Mover Wizard (click Help).

5. Read the welcome screen and click Next. The Select your source files page opens.

   Select your source files

6. Verify that the correct Metadata Repository Source is specified.
The metadata repository source is a Microsoft Access database that was installed as part of the Enterprise Manager installation. If it is not in the default Enterprise Manager directory, search for EM_Proj_MD.mdb and type the correct path in the field.

7 Verify that the correct Warehouse Source is specified.

The warehouse source is also a Microsoft Access database that was installed as part of the Enterprise Manager installation. If it is not in the default Enterprise Manager directory, search for EM_WH.mdb and type the correct path in the field.

8 Leave the Run a SQL script check box selected, and make sure that the Overwrite warehouse tables if they exist check box is selected.

9 Click Next. The Project Selection page opens.

10 Select Enterprise Manager and click Next. The Run a SQL Script on the Warehouse page opens.

11 In the SQL script to be run after warehouse transfer field, specify the Enterprise Manager SQL script appropriate to your data warehouse:

a Click the browse (...) button. A window opens to a directory containing .sql files.

b Select the appropriate em_sql_databasetype.sql file for your Enterprise Manager data warehouse:

If you use the Enterprise Manager warehouse in Oracle 9i or higher, use the em_sql_ora_std_ed.sql script. This script...
uses SQL-92 syntax, and results in better execution and improved performance for Enterprise Manager queries.

<table>
<thead>
<tr>
<th>Data warehouse</th>
<th>SQL script to run</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server 2003 or earlier</td>
<td>em_sql.sql</td>
</tr>
<tr>
<td>Oracle 8i</td>
<td>em_sql_ora.sql</td>
</tr>
<tr>
<td>Oracle 9i or newer</td>
<td>em_sql_ora_std_ed.sql</td>
</tr>
<tr>
<td>DB2</td>
<td>em_sql_db2.sql</td>
</tr>
<tr>
<td>Teradata</td>
<td>em_sql_tera.sql</td>
</tr>
<tr>
<td>Sybase</td>
<td>em_sql_syb.sql</td>
</tr>
</tbody>
</table>

12 Click **Next**. The Metadata Repository Location page opens.

13 Select the **Data Source Name** for the Enterprise Manager metadata database from the drop-down list.

14 Type a valid **User Name** and **Password** for this database.
15 Click **Next**. The Warehouse Location page opens.

![Warehouse Location](image)

**Warehouse Location**

16 Select the **Data Source Name** for the Enterprise Manager data warehouse from the drop-down list.

17 Type a valid **User Name** and **Password** for this database.

18 Select the correct database type from the **Database Connection Type** drop-down list.

19 Click **Next**. The Metadata Repository Connection page opens.

![Metadata Repository Connection](image)

**Metadata Repository Connection**

20 In the **Project Source Name** field, type the name of the project source to create the Enterprise Manager project in. If this project source does not exist, it will be created.
21 Click **Next**. The Summary page opens.

22 Review the information on the summary page and click **Transfer** when you are ready. The process can take several minutes.

23 When the transfer process is complete, the Project Mover Wizard closes and you are returned to the Enterprise Manager Console. Click **Next**. The Configure Overview page opens.

**Selecting the projects to monitor**

In this part of the Enterprise Manager Console, you determine which projects to monitor using Enterprise Manager. These projects must be configured to log statistics.

---

**To select which projects to monitor in Enterprise Manager**

1. Read the Configure Overview page and click **Next**. The Choose Projects page opens.

2. From the **Available Server Project Sources** list, select the project source that contains the projects that you want to monitor using Enterprise Manager.

   If the project source that you want to monitor has the default name of “New Project Source,” you must change the name of the project source before adding it to Enterprise Manager. Otherwise Enterprise Manager will be unable to load statistics and fact data from the projects in the project source.
3 Click the right arrow (>). The Select Projects (Project Source Name) dialog box opens.

4 In the User Login area, type a MicroStrategy login ID that has Administrator access to the selected projects, and the password for that login ID.

Enterprise Manager console only supports standard authentication. Warehouse and single sign-on authentication modes are not supported. For information about the different authentication modes, see Chapter 3, Identifying Users: Authentication.

5 Click Validate. The available projects are listed in the selection area.

6 Select the projects you want to monitor with Enterprise Manager and click OK.

You can only select projects that have had statistics logging enabled.
7 Repeat steps 2 through 6 for any other project sources containing projects you want to monitor.

In a clustered environment, you can only monitor a project once, regardless of how many Intelligence Servers it is running on. This is because Enterprise Manager connects to the project to populate the lookup tables, and because the project is the same on all Intelligence Servers when clustered. All of the statistics are available in Enterprise Manager as long as each clustered Intelligence Server is set up to log statistics to the Enterprise Manager data warehouse.

8 Click Next. The Schedule Overview page opens.

Scheduling how often statistics are loaded into the Enterprise Manager data warehouse

This procedure configures how often data is automatically loaded from the statistics tables into the Enterprise Manager data warehouse. For detailed information about the data load process, see Data loading, page 660.

The data load process is scheduled to run automatically, as a Windows service. To manually force a load, click Run Manual Loading Now.

You can also schedule the data load process to run by using the Windows AT command and calling the MicroStrategy Enterprise Manager data load command: maemetl. For syntax information, see To perform Enterprise Manager maintenance tasks from the command line, page 666.

To set the schedule for Enterprise Manager data loading

1 Read the Schedule Overview Page and click Next. The Define Schedule page opens.

Recurrence

2 Select either Daily, Weekly, or Monthly. Depending on which you select, the options to the right change. Then specify the days on which you want the data load to occur.
**Time Range**

3 To load data once per day, select **Occurs once at** and specify the time of day when you want the data load to start.

4 To load data several times a day:
   a Select **Occurs every** and specify the interval between data loads, in minutes or hours.
      This interval must be longer than the time taken by the data load. For example, if the data loading process takes 10 minutes to run, set a time interval longer than 10 minutes
   b Select the time of day at which you want the data load process to start.
   c Select the time of day after which the data load process does not start.

**Date Range**

5 Select the date on which to start the scheduled data load.

6 To run the data load process indefinitely, select **No end date**.

7 To specify a date after which no data is loaded into the Enterprise Manager data warehouse, select **End by** and specify a date.

**Maintenance tasks**

8 From the **Tools** menu, select **Options**. The Options dialog box opens.

9 Select the **Data Load** tab.

10 Select the check box for any maintenance tasks you want to perform during each data load. For a description of each task, see *Enterprise Manager maintenance tasks, page 665*.

11 Click **Apply Changes**.

   Changes made on this page do not take effect until you click **Apply Changes**.

12 To restart the data loading service with the new schedule, click **Stop** to stop it, and then click **Start**.

To load statistics immediately and not wait for the first scheduled load to occur, click **Run Manual Loading Now**.
To trigger an immediate data load from a Windows Command Prompt, use the `maemetl` command. For syntax information, see "To perform Enterprise Manager maintenance tasks from the command line, page 666."

**Enterprise Manager Setup Complete**

The final page of the Enterprise Manager Console provides a link to open MicroStrategy Desktop. Once in MicroStrategy Desktop or Web, you can log in to the Enterprise Manager project and start running reports as in any other MicroStrategy project.

The Enterprise Manager metadata contains a user group called EMAdmin. Users in this group have all the privileges necessary to use all features of Enterprise Manager. It is recommended that any users who are to administer Enterprise Manager be added to this group.

**Upgrading Enterprise Manager**

There are several important issues to consider as you upgrade your statistics database, Enterprise Manager warehouse, and Enterprise Manager project. See the *MicroStrategy Upgrade Guide* for complete details on upgrading your Enterprise Manager environment.

The best way to update the Enterprise Manager project is by using the Project Merge Wizard, as described in the *Upgrade Guide*. Alternately, you can use Object Manager to upgrade the Enterprise Manager project. For a procedure for upgrading Enterprise Manager with Object Manager, see MicroStrategy Tech Note TN15976. For instructions on how to use Object Manager and Project Merge, see *Chapter 5, Managing Your Projects*.

**Maintaining Enterprise Manager**

This section provides information about the following topics:

- *Data loading, page 660*
- *Enterprise Manager maintenance tasks, page 665*
- *Improving warehouse performance by purging unused data, page 668*
Data loading

The statistics tables contain raw data logged by Intelligence Server. For performance reasons, most fields are fixed length fields and contain data that cannot be directly interpreted by an administrator. An example of such a field is an object GUID. This raw data must be processed further to support administrative analysis and reporting requirements. This processing is called data loading. For a diagram of the data loading process, see *Understanding the Enterprise Manager architecture*, page 641.

The first half of the Enterprise Manager data load process involves gathering metadata from projects. The system connects to the project sources specified in the Enterprise Manager console and transfers relevant information to the Enterprise Manager lookup tables. Relevant information includes such data as report names, user/group names, and object relationships. Examples include user/group relationships and which schedules are mapped to which reports.

Metadata information for all projects in a given project source is transferred into the Enterprise Manager lookup tables, regardless of whether those projects are configured to log statistics.

The second half of the data load process involves transferring facts from the statistics tables into the Enterprise Manager warehouse. SQL scripts transform the statistics data into a form that can be useful for administrative reporting in the Enterprise Manager warehouse. The transformation ensures that reporting on MicroStrategy metadata content is feasible.

Some of Enterprise Manager's fact tables are really views of certain statistics tables. This substantially speeds up the data load process.

To ensure that the warehouse data is complete, at the beginning of the data load process a timestamp is created in the EM_IS_LAST_UPDATE table, according to the current date and time in the Enterprise Manager data warehouse. This timestamp provides the end of the data migration window. The beginning of the data migration window is determined by the previous data load’s timestamp entry in the EM_IS_LAST_UPDATE table. Therefore, any statistics logged between the start of the last data load and the start of the current data load is transferred. At the completion of the data load process, Enterprise Manager updates the EM_IS_LAST_UPDATE table to show that the process is finished.

If the data load process is interrupted before it finishes, this last update is not stamped. In this case, the next time the data load runs, it starts with data from the time the last successful data load was finished.
Data loading generally occurs according to the schedule that you set up when you configure Enterprise Manager. For instructions on changing the schedule for the data load, see *Scheduling how often statistics are loaded into the Enterprise Manager data warehouse, page 657*. You can also run a manual data load at any time, or manually load data from a specific time window if your warehouse is missing data for some reason.

For information about troubleshooting the Enterprise Manager data loading process, see *Troubleshooting the data loading process, page 664*.

**Best practices for Enterprise Manager data loading**

- Set up the scheduled data loads based on the answers to these questions:
  - How long does the data load take?
  - How current does the data need to be?

  If you need near-real-time data, and the data load does not take longer than a few minutes, you can run the data load as often as once per hour. However, if the data load process takes a long time, you should run it during times when Intelligence Server usage is low, such as overnight.

- The data load maintenance tasks may significantly impact the load on your Intelligence Server. **Close open sessions**, for example, requires very little overhead. MicroStrategy recommends that you run this task with every data load. In contrast, **Update database statistics** can increase the load dramatically. If you are running frequent data loads, this task should not be enabled by default during all data loads. For more information about the data load options, see *Enterprise Manager maintenance tasks, page 665*.

- When you delete a monitored project in Desktop, you must also remove it from the Enterprise Manager console. Otherwise Enterprise Manager will attempt to load data from that project during data load.

- When you change the name of a project in Desktop, you should refresh the list of available projects in the Enterprise Manager console before the next data load. Otherwise Enterprise Manager will continue to refer to the project under its old name.

- Synchronize the time of the Intelligence Server machine with the data warehouse time if possible. When Intelligence Server writes statistics into the database, it uses the data warehouse RDBMS timestamp. This is written as `em_record_ts` (in the fact tables) and as `recordtime` (in the statistics tables). Enterprise Manager uses `recordtime` to determine which statistics to move over according to the “time window” for a data
load process. The time window is determined by the data warehouse RDBMS time.

Also, if the data warehouse is different from the Intelligence Server machine time, certain reports in Enterprise Manager can have missing data. 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.

---

To change the data loading schedule

1. In the Enterprise Manager console, select **Schedule**.
2. Click **Next**.
3. Make any necessary changes to the **Time Range**, **Recurrence**, or **Date Range** options. For details about each option, see *Scheduling how often statistics are loaded into the Enterprise Manager data warehouse, page 657*, or click **Help**.
4. Click **Apply Changes**.
5. To restart the data loading service with the new schedule, click **Stop** to stop it, and then click **Start**.

Changes made on this screen do not take effect until you click **Apply Changes** and restart the data loading service.

---

To run a manual data load immediately

1. In the Enterprise Manager console, select **Schedule**.
2. Read the Scheduling Overview page and click **Next**.
3. Click **Run Manual Loading Now**. A dialog box opens with a progress bar indicating the progress of the data load process. When the data load finishes, click **OK** to close this dialog box.
To load data for a specific time period

1. In the Enterprise Manager console, from the Tools menu, select Data Load Options.

2. In the drop-down list boxes, specify the Start Date and End Date for the custom time window.

3. Click Custom Time Window. The Data Load Options dialog box closes, and a dialog box opens with a progress bar indicating the progress of the data load process. When the data load finishes, click OK to close this dialog box.

To execute a data load from the command line

Call the Data Loader executable using the command `maemetl.exe`, with one or more of the parameters listed below. For example, to execute a data load, the syntax is

```
maemetl.exe -dm
```

The `maemetl.exe` file is located by default in the following location:

```
C:\Program Files\MicroStrategy\Enterprise Manager
```

The data loading parameters are as follows:

- `-DM`: Data migration process. This switch migrates data from the statistics tables to the Enterprise Manager data warehouse tables.

- `-CWM -MM/DD/YYYY -MM/DD/YYYY`: Custom time window migration. This option loads all statistics data logged between the specified dates.

You can call the Data Loader executable with data load parameters, maintenance parameters, and warehouse purge parameters at the same time. For a list of maintenance parameters, see To perform Enterprise Manager maintenance tasks from the command line, page 666. For a list of warehouse purge parameters, see To purge Enterprise Manager warehouse data from the command line, page 668.
Troubleshooting the data loading process

Enterprise Manager logs all the information from the data load process, including all errors, in the MSTRMigration.log text file. By default this file is stored in:

```
C:\Program Files\MicroStrategy\Administrator\Enterprise Manager\
```

If an error occurs during the execution of a SQL script, the SQL script that caused the error is recorded in the log file. The script can then be run using a native layer query tool against the Enterprise Manager warehouse. This can identify whether the error is due to an error in the SQL script, or to missing or incorrect statistics data.

Below are some of the possible reasons why a data load might fail:

- Data cannot be loaded from a project that is not currently loaded on Intelligence Server. Data also cannot be loaded from a project that is set to Request Idle, Execution Idle, or Full Idle mode.

  Before loading data from a project, make sure it is not in any of these idle modes and is set to Loaded status. For an explanation of the different project modes, including instructions on how to set a project’s mode, see Setting the status of a project, page 31.

- If you have changed the password for the user that configured a project for Enterprise Manager, data cannot be loaded from that project until you reconfigure it in the Enterprise Manager console. In the 2. Configure step, remove the project source from the list of project sources being monitored, and then add it to the list again.

- If you have deleted a project in Desktop that is being monitored by Enterprise Manager, the data load process fails until you remove that project from the list of projects being monitored.

- If the MicroStrategy Enterprise Manager Data Loader service has stopped, the scheduled data loads will not occur. Restart the Data Loader service to resume scheduled data loading.

For additional possible causes of data load failure, see MicroStrategy Tech Note TN16852.
Enterprise Manager maintenance tasks

In addition to loading data from the statistics tables and project metadata, the data load process can perform certain system maintenance tasks. These tasks keep your Enterprise Manager project and data loads performing efficiently. They can be performed during each data load, or can be run immediately.

The maintenance tasks that can be performed are:

- **Update folder paths**: This task updates the location property of attributes such as Report, User, and so on. It synchronizes the Enterprise Manager warehouse lookup tables with the actual folder paths in the metadata.

- **Update object deletions**: This task ensures that objects that are deleted in the project metadata are marked as having been deleted in Enterprise Manager. Information about deleted objects is retained in the Enterprise Manager lookup tables for historical analysis. A deleted object is marked with a Deleted flag in the corresponding lookup table.

- **Update database statistics**: This task executes SQL scripts that cause the Enterprise Manager warehouse and statistics database to collect statistics on these warehouse tables. The database uses these statistics to improve response times for Enterprise Manager reports. This option is available for SQL Server, Oracle, Teradata, and DB2 version 8.2.2 or later. This task should be run frequently to improve the performance of Enterprise Manager reports.

  The SQL scripts that are run for this option are:

  - `Upd_Stat_Table_Stats_DBname.sql`
  - `Upd_Fact_Table_Stats_DBname.sql`

- **Close open sessions**: This task closes all sessions that are listed as open in the statistics database. Using this task helps to avoid orphan sessions, entries in the statistics database that indicate that a session was initiated in Intelligence Server, but no information was recorded when the session ended. Orphan sessions occur rarely, but they can affect the accuracy of Enterprise Manager reports that use Session Duration. For example, one long-running orphan session may skew the average time a session lasts by several days. When this task is executed, all sessions whose disconnect time is NULL that have been open for more than 24 hours have their disconnect time updated to 24 hours after the connect time.

  The SQL script run for this option is `em_close_orphan_sessions_DBname.sql`.
• **Repopulate relate tables**: This task synchronizes the relationship (relate) tables in the Enterprise Manager warehouse, such as IS_SCHED_RELATE or IS_USR_GP_USR, with the metadata.

Since the maintenance tasks are performed during a data load, for troubleshooting suggestions regarding these tasks, see *Troubleshooting the data loading process, page 664.*

---

**To select the maintenance tasks to be performed during each data load**

1. From the **Tools** menu in the Enterprise Manager Console, select **Options**. The Options dialog box opens.
2. Select the **Data Load** tab.
3. Select the tasks to perform during the data load. For a detailed explanation of each task, see *Enterprise Manager maintenance tasks, page 665.*
4. Click **OK** to save your choices and close the Options dialog box. The selected data load options will be performed during each subsequent data load.

---

**To perform an Enterprise Manager maintenance task immediately**

1. From the **Tools** menu in the Enterprise Manager Console, select **Data Load Options**. The Data Load Options dialog box opens.
2. Click the button corresponding to the task to perform. For a detailed explanation of each task, see *Enterprise Manager maintenance tasks, page 665.*

---

**To perform Enterprise Manager maintenance tasks from the command line**

Call the Data Loader executable using the command `maemetl.exe`, with one or more of the parameters listed below. For example, to update object locations and clean up orphan sessions, the syntax is

```
maemetl.exe -ol -cos
```

The `maemetl.exe` file is located by default in the following location:
The maintenance task parameters are as follows:

- **–OL**: Object location update process. This switch updates the Location property of all objects. If users have moved objects or changed their folder structure, this switch allows Enterprise Manager to be synchronized with the users' project structure. This is the same as the **Update Folder Paths** option.

- **–EF**: Exists flag update process. This switch determines which objects monitored by Enterprise Manager no longer exist in users' projects. This is the same as the **Update Object Deletions** option.

- **–UWTS**: Update Enterprise Manager warehouse table statistics process. This switch helps improve the query performance on the database. The database's optimizer uses these statistics when running SQL. This switch is available for SQL Server, Oracle, Teradata, and DB2 version 8.2.2 or later. This option is part of the **Update Database Statistics** option.

- **–USTS**: Update Intelligence Server statistics table statistics process. This switch helps improve the query performance on the database. The database's optimizer uses these statistics when running SQL. This switch is available for SQL Server, Oracle, Teradata, and DB2 version 8.2.2 or later. This option is part of the **Update Database Statistics** option.

- **–COS**: Close orphan sessions process. This switch updates the information in users' statistics table to close any sessions left open by Intelligence Server for any reason. This is the same as the **Close Orphan Sessions** option.

- **–REP**: Repopulate Enterprise Manager relate tables process. This option synchronizes the relationship tables with the metadata as of the time of the data load. This is the same as the **Repopulate Relate Tables** option.

You can call the Data Loader executable with data load parameters, maintenance parameters, and warehouse purge parameters at the same time. For a list of data load parameters, see *To execute a data load from the command line, page 663*. For a list of warehouse purge parameters, see *To purge Enterprise Manager warehouse data from the command line, page 668*. 
Improving warehouse performance by purging unused data

Enterprise Manager collects a great deal of data. Over time some of this data may become outdated and no longer relevant to your analysis. When this happens, MicroStrategy recommends that you purge the outdated data from the data warehouse to ensure optimum database performance. Purging unused or out-of-date data from the warehouse keeps the database to a manageable size, making it easier to maintain and faster to query.

Warehouse data deletions are not logged in the EM_LOG database table.

All database purge processes are run as a database transaction, and the transaction is rolled back in the event of any failure, to ensure that the statistics database is in a consistent state. To keep the size of these transactions small, it is recommended that you set many small time windows for purge operations rather than specifying a single large time window, depending on the size of your statistics database. For example, for a very large database, the time window can be reasonably set to one month; for a small database a more appropriate time span might be one year.

To purge data for a specified time period from the Enterprise Manager warehouse

1 On the Enterprise Manager console, from the Tools menu select Data Load Options. The Data Load Options dialog box opens.

2 In the Specify custom time window area, set a Start date and End date.

3 Click WH data deletion. The Purge Projects dialog box opens.

4 From the list of projects, select the projects whose data you want to purge from the warehouse.

5 Click OK. All data pertaining to the selected projects during the specified time window is deleted from the Enterprise Manager warehouse.

To purge Enterprise Manager warehouse data from the command line

Call the Data Loader executable using the command maemetl.exe, with the parameter 

\[-WHD -MM/DD/YYYY -MM/DD/YYYY [-S"ServerName" -P"ProjectName"]\]. You can choose to purge data for only a specific server, or for only a specific project on a specific
server. For example, to purge the warehouse data for all projects on the Intelligence Server machine ARCHIMEDES between January 1 and March 31, 2008, the syntax is:

```
maemetl.exe -whd -01/01/2008 -03/31/2008 -S"ARCHIMEDES"
```

The `maemetl.exe` file is located by default in the following location:

```
C:\Program Files\MicroStrategy\Enterprise Manager
```

You can call the Data Loader executable with data load parameters, maintenance parameters, and warehouse purge parameters at the same time. For a list of data load parameters, see *To execute a data load from the command line, page 663*. For a list of maintenance parameters, see *To perform Enterprise Manager maintenance tasks from the command line, page 666*.

---

**Reporting in Enterprise Manager**

Enterprise Manager contains many reports designed to provide you with useful information about your MicroStrategy objects, report processing data, and user and session data. For example, is the length of time jobs wait in queue causing significant delays in report processing? If so, increasing the number of available database connection threads may help decrease the queue time. This is one example of how Enterprise Manager reports can help you.

Enterprise Manager also contains a number of dashboards, a type of interactive document which uses one or more reports to explore related areas of data. Make use of dashboards before you begin executing any reports for an intuitive introduction to the ways in which Enterprise Manager can help you analyze report information. For more details on the contents of the dashboards included with Enterprise Manager, see *Dashboards, page 673*.

To run a report included in the Enterprise Manager project, you connect to it as you would any of your other projects. You must have been assigned the appropriate privileges in the Enterprise Manager project by a system administrator.

You can use the Enterprise Manager reports out of the box, or you can modify the reports to return exactly the data you want to analyze for your MicroStrategy environment. For information about customizing the
Enterprise Manager reports, see *Customizing Enterprise Manager reports to suit your needs, page 672*.

Enterprise Manager report names are preceded by a number. The integer-numbered reports (such as 81. Activity by User) indicate that the report returns data on a high-level analysis area. The decimal-numbered reports (such as 81.1 Ad-hoc Job Activity by User or 81.2 DB Result Rows by User) indicate that these reports provide more detail within the higher-level analysis area.

Indexes are included for out-of-the-box reports on the Enterprise Manager fact tables. Check the indexes for the Enterprise Manager reports that you run most frequently or that take the longest to complete. If necessary, you should build additional indexes if you find some reports using tables that do not currently have an index.

The analysis areas of the Enterprise Manager project are described below. Several of the analysis area descriptions include details on one or two representative reports from that area, and report customization ideas that can be used with many of the reports within that analysis area.

- **Dashboards** are an excellent source of summarized data, and provide interactive analysis at deeper levels of detail. For descriptions of each Enterprise Manager dashboard, see *Dashboards, page 673*.

- **Operations analysis** reports provide information on system resource usage, concurrency, and report and subscription processing time. For descriptions of these reports, see *Operations analysis, page 684*.

- **Performance analysis** reports support analysis related to usage patterns, Intelligence Server governing settings, and Intelligent Cube Analysis. For descriptions of these reports, see *Performance analysis, page 693*.

- **Project analysis** reports provide information about MicroStrategy project growth and the usage patterns of configuration and project objects. For descriptions of these reports, see *Project analysis, page 697*.

- **Real-time analysis** reports provide information related to current response times and schedule results. This information can be useful for troubleshooting and for optimizing your database configuration. For descriptions of these reports, see *Real-time analysis, page 705*.

- **User analysis** reports analyze user activity and preferences. For descriptions of these reports, see *User analysis, page 706*.
For a detailed list of all Enterprise Manager facts, attributes, and metrics, see the Enterprise Manager Data Dictionary in the Supplemental Reference for System Administration.

Since Intelligence Server can be configured to log different types of statistics, some of the reports in the Enterprise Manager project can be affected if certain logging options are not selected. For detailed information about the statistics logging options, see Monitoring system usage: Intelligence Server statistics, page 323.

**Best practices for Enterprise Manager reporting**

- While a recent Enterprise Manager data load is not a prerequisite to viewing and aggregating recently collected statistics data, to view object names from the project metadata a prior successful data load must have occurred in which the object names and descriptions were loaded. If this has not occurred, metrics are still reported and aggregated correctly but certain object names appear as null fields.

  Once the statistics data go through the data loader, the scope of analysis increases significantly and reports in the various Enterprise Manager analysis areas will reflect the data.

- The dashboard documents included with Enterprise Manager give an intuitive introduction to reporting and should be used prior to running any reports. For more information on the dashboards available in Enterprise Manager, see Dashboards, page 673.

- Instead of directly modifying a report supplied with Enterprise Manager, MicroStrategy recommends that you make a copy of the report and modify that copy.

- By default, only the MicroStrategy system administrator and users in the MicroStrategy Web Viewer group have the necessary permissions and privileges to run reports in Enterprise Manager. If other users need to view the Enterprise Manager reports, an administrator must assign them the privileges to use all objects in the Public Objects and Schema Objects folders.

- Data that is displayed as “########” indicates that the data cannot be displayed. For example, if a metric for Average Report Execution Duration displays “########” for a report within a certain time frame, that report may have never executed to completion during that time frame, and had all its executions cancelled. This may mean that you need to investigate your system for problems during that time frame.
Customizing Enterprise Manager reports to suit your needs

The reports included with Enterprise Manager provide a wide variety of analysis. However, you may find that the out-of-the-box reports do not fit your specific analytical needs. In this case, you can either edit an existing report to fit your needs, or you can create your own report using the predefined metrics and attributes in the Enterprise Manager project.

Instead of directly modifying a report supplied with Enterprise Manager, MicroStrategy recommends that you make a copy of the report and modify that copy.

You can modify and create reports in Enterprise Manager through MicroStrategy Desktop or MicroStrategy Web, in the same way as in any other MicroStrategy project. For an introduction to MicroStrategy reporting, covering the basics of analyzing and creating reports and report objects such as metrics and filters, see the MicroStrategy Basic Reporting Guide. For information about more complex reporting tasks, see the MicroStrategy Advanced Reporting Guide. You can also click Help for detailed information about any interface.

Some Enterprise Manager reports make use of MicroStrategy’s Freeform SQL feature to allow you to use custom SQL statements to access alternative data sources. For complete details about Freeform SQL, see the MicroStrategy Advanced Reporting Guide.

Viewing information about Enterprise Manager objects:

Project documentation

The Project Documentation Wizard can provide detailed information about any objects in the Enterprise Manager project. For example, to find which metric to use for a specific purpose, you can generate and view the project documentation for the Enterprise Manager metrics.

When you step through the Project Documentation Wizard, you specify the objects that want information about. The Project Documentation Wizard records information about these objects in an HTML file. You can view and search this HTML file, or print it for quick reference later.

The Enterprise Manager project documentation is initially only available on the machine that has the Enterprise Manager project installed. To enable other users to view the project documentation, share the HTML files over your network.
To create project documentation for Enterprise Manager

1 In Desktop, log in to the Enterprise Manager project. You must log in with an account that has administrative privileges in this project.

2 From the Tools menu, select Project Documentation. The Project Documentation Wizard opens.

3 Step through the Project Documentation Wizard. Select the Enterprise Manager project, and the objects and information that you want to include in the project documentation. For instructions on how to use the wizard, click Help.

To include information about metrics or reports, select Application objects. To include information about attributes, select Schema objects.

4 On the last page of the wizard, click Finish. The project documentation HTML files are generated in the location you specify in the wizard.

Dashboards

Enterprise Manager comes with several Report Services documents that show one or more related reports in a dashboard-type display. Report Services documents are an excellent source of summarized data from related areas of analysis. Dashboards, which are a type of document, provide a lot of interactive graphical features to enable exploration of the data at several levels of detail.

The Enterprise Manager Overview dashboard provides an intuitive and interactive overview of collected document, report, and user activity data, as well as a quick guide to the other dashboards available in Enterprise Manager. Make use of this dashboard before executing any Enterprise Manager reports to obtain an excellent introduction to the capabilities of Enterprise Manager.

You must have a MicroStrategy Report Services license to view or work with a Report Services document. In addition, dashboards must be viewed in MicroStrategy Web to take full advantage of their interactivity.
The other dashboards provided with Enterprise Manager are:

- **Data Warehouse Optimization Advisor, page 674**
- **Document Processing Analysis Dashboard, page 675**
- **Project Analysis Dashboard, page 676**
- **Project Analysis Interactive Dashboard, page 678**
- **Real-Time Server Usage Dashboard, page 679**
- **Report Processing Analysis Dashboard, page 681**
- **Server Caching Optimization Advisor, page 682**
- **User Analysis Dashboard, page 683**

These dashboards are designed for use with MicroStrategy Web. If you are using MicroStrategy Web Universal, the links to other reports in the dashboards do not function. To correct the links, edit the dashboards and change all occurrences of `Main.aspx` in the links to `mstrWeb`.

**Data Warehouse Optimization Advisor**

This dashboard provides information that can guide you when optimizing the performance of your data warehouse. The user specifies the time frame for analysis, the number of database tables to consider for optimization, and the percentage of jobs to consider in the optimization calculations.
The top half of the dashboard summarizes the current database usage. This summary includes:

- An executive summary of the database usage over the specified time period, including the number of report requests and report result rows, the average execution time for each request, and the total execution time.

- A gauge graph showing the percent of total database load that is taken up by the specified percentage of reports.

- A grid showing which reports consume the most database resources. This grid can be sorted by number of report requests, number of result rows, or execution time.

The bottom half of the dashboard contains a list of the database tables being considered for optimization. Clicking on a table brings up a list of optimizations and their potential effectiveness. These optimizations include aggregate table grouping and different types of secondary indexes.

### Document Processing Analysis Dashboard

This dashboard provides an overview of document usage and performance in your projects, including average wait times and number of execution errors encountered. The user specifies the time period for analysis.
The Processing Summary section provides a general picture of document activity for a given project and Intelligence Server. It includes:

- Line graphs showing key performance indicators for document execution, including average wait times, execution times, and number of errors encountered.
- An area graph showing the number of jobs and execution errors over time.
- An area graph showing average wait, execution, and queue times for executed documents.

The Document Details section provides in-depth information on specific documents within a given project. Select a document from the drop-down list to view:

- A bubble graph of popular documents, with the number of users and job requests for each document.
- General information about the selected document, including the owner, description, and date the document was last modified.
- An area graph showing the average wait time, execution time, and queue time encountered when executing the document.

**Project Analysis Dashboard**

This dashboard provides a comprehensive overview of usage and activity on your projects, over the time frame that you specify.
The top portion of the dashboard contains a general analysis of the system. It includes:

- Bar graphs showing the project usage by total time spent executing jobs, total number of jobs executed, and number of users connected.
- An overview of system usage for each project, including cache hits, number of jobs, and other metrics.

On the dashboard, below the general analysis section is a separate section for each project. These sections contain a detailed analysis of the project, including:

- A line graph showing the weekly growth trend for the numbers of reports and other objects in the project.
- A line graph showing the weekly usage trend, in terms of number of users and number of user requests.
- A line graph showing the weekly project performance trend, in terms of job execution time and number of jobs.
- A graph showing the load distribution (ad-hoc versus scheduled jobs).

This dashboard also contains a number of links to other Enterprise Manager reports.

For customization purposes, the document links work over the ASP.NET version of MicroStrategy Web. For MicroStrategy Web Universal, the links must be modified appropriately. The MicroStrategy Developer Library (MSDL) that comes with the MicroStrategy SDK provides information to customize Report Services documents.
Project Analysis Interactive Dashboard

This dashboard provides usage information for a given project, including object storage, patterns in project growth, and a list of unused application objects for cleanup.

The Overview section provides a general view of project performance and project size. It includes:

- Line graphs of key performance indicators for the project over time, including the number of users, sessions, and document jobs.

- Bar graphs of the number of application and schema objects in the project, broken down by object type. These include reports, documents, attributes, and hierarchies.

The Growth section provides details about new application objects added to the project. It includes:

- A graph of the number of application objects that have been added to the project by week, broken down by object type.

- A funnel graph of new application objects added to the project, organized by the owner of the object.
The Usage section provides a picture of the most frequently used objects in the project. It includes:

- Heat maps of the top ten frequently used reports and documents in the project.
- Lists of the top ten attributes, metrics, and filters in the project.

The Clean Up section provides a view of unused objects in the project, including:

- A list of unused reports, documents, attributes, and other application objects for the project.
- Links to generate scripts for deleting unused application objects from the metadata.

**Real-Time Server Usage Dashboard**

This dashboard provides an overview of recent system activity. The user specifies the earliest date for which data is reported.
The dashboard includes:

- Critical system metrics such as average job duration and total jobs processed for a given server machine.
- Pie charts indicating the recent job distribution by project, and execution status such as Completed, Error, and so on.
- A line graph showing system throughput and response time for a given server machine.
- A grid indicating the current status of scheduled jobs on the server machine, on a per project basis.

The dashboard also includes links to more detailed reports.

For customization purposes, the document links work over the ASP.NET version of MicroStrategy Web. For MicroStrategy Web Universal, the links must be modified appropriately. The MicroStrategy Developer Library (MSDL) that comes with the MicroStrategy SDK provides information to customize Report Services documents.
**Report Processing Analysis Dashboard**

This dashboard provides an overview of report usage and performance in your projects, including average wait times and number of execution errors encountered. The user specifies the time period for analysis.

**The Processing Summary section** provides a general picture of report activity for a given project and Intelligence Server. It includes:

- Line graphs of various key performance indicators, including the number of user report requests, exported jobs, and execution times.
- An area graph showing the number of jobs and report execution errors encountered over time.
- A graph showing average wait times, execution times, and queue times for reports over time.

**The Reporting Details section** provides in-depth information on individual reports within a given project. Select a report from the drop-down list to view:

- A bubble graph of popular reports, with the number of user and job requests for each report.
- Detailed information about the specified report, including the owner, description, and date the report was last modified.
- A heat map showing the users that have requested the report, along with the number of report execution errors encountered.

**Server Caching Optimization Advisor**

This dashboard provides information that can guide you towards optimal caching strategies, based on historical usage data. The user specifies the percentage of jobs to be optimized (the default is 20%) and a period of time for the analysis.

The dashboard includes two panels. The first panel provides an overview of the effectiveness of server caching over the specified time period. It contains:

- A summary of the server workload, including the number of report requests, the average amount of time required to execute a request, the average number of result rows, and the number of jobs that resulted in an error.
- A gauge showing the percentage that database execution time has been reduced by jobs that hit the cache instead of the database.
- A bar graph analyzing the hourly server workload by average time each job spends in queue, average execution time per job, and number of jobs per hour.
The second panel provides the optimization potential for three different optimization strategies, presented in three different grids:

- Enabling caching for the worst-performing reports, based on projected database savings.
- Disabling caching for reports with low hit ratios.
- Increasing caching efficiency by building OLAP cubes augmented with frequently drilled-to objects.

User Analysis Dashboard

This dashboard provides an overview of user activity on an Intelligence Server or project, including statistics for report and document execution and user session information. You can also examine inactive users to determine how long they have been inactive or determine whether they should be removed.

The Intelligence Server Usage section shows user and session activity for a given Intelligence Server. It includes:

- Line graphs of key performance metrics for Intelligence Server. This includes average session durations, average connection times, and number of distinct users.
• A heat map showing the number of users connecting to Intelligence Server via applications such as MicroStrategy Web or MicroStrategy Scheduler.

• Area graphs with breakdowns of the number of sessions, number of users, and average session durations per application or project.

The Project Usage section contains user-specific activity information for a given project. Select a project name from the drop-down list to view:

• A heat map of connection activity by user, with number of jobs and sessions per user and connection times.

• A breakdown of each user’s individual activity, including the names of executed reports and documents, errors encountered, and wait times.

The User Inactivity section allows you to view information on inactive users by project or Intelligence Server, including:

• An area map of inactivity trends showing the number of users that have been inactive for a given period of time.

• A list of inactive users, with a listing of account creation dates and date of last connection.

• A bar graph showing the distribution of inactive users by length of inactivity.

**Operations analysis**

The Operations Analysis folder in Enterprise Manager contains the following analysis areas, each with its own reports:

• *Concurrency analysis (including user/session analysis), page 685*

• *Data load, page 686*

• *Delivery processing analysis, page 687*

• *Inbox Message Analysis, page 687*

• *Report processing analysis, page 688*

• *Resource utilization analysis (including top consumers), page 690*
Concurrency analysis (including user/session analysis)

This analysis area provides reports to analyze session and user concurrency on the system at different points in time. Administrators can measure:

- The total number of users in the metadata
- The total number of users that are currently logged in
- The total number of user sessions that are currently open
- The total number of user sessions that have active jobs

Reports in this analysis area use an attribute from the Time hierarchy as the primary attribute for analysis, as well as various metrics representing answers to administrator questions.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. Concurrency by Hour of Day</td>
<td>Provides the number of concurrent active users and the number of concurrent sessions by hour of day. This report is prompted on time.</td>
</tr>
<tr>
<td>11. Daily Session Concurrency Analysis</td>
<td>Uses various metrics to analyze the concurrent active sessions over time. This report is prompted on time and on session duration.</td>
</tr>
<tr>
<td>12. Session Duration Analysis</td>
<td>Uses various metrics to analyze the duration of user sessions, over time. This report is prompted on time and on session duration.</td>
</tr>
<tr>
<td>13. Daily User Connection Concurrency Analysis</td>
<td>Uses various metrics to analyze the concurrency of user sessions, over time. This report is prompted on time and on session duration.</td>
</tr>
<tr>
<td>14. Minute Level User Concurrency During Peak Hours</td>
<td>Provides a minute-level graph for the active users and sessions during the peak hours of the day. This report is prompted on time.</td>
</tr>
<tr>
<td>14.1 Top n Maximum User Concurrency Hours - report as filter</td>
<td>Provides a list of the top N hours in terms of maximum user concurrency. This report is prompted on time, session duration, and number of hours to be returned.</td>
</tr>
</tbody>
</table>

Sample report: Daily Session Concurrency Analysis

This report uses various metrics to analyze the concurrent active sessions over time. The results summarize the load on and usage of Intelligence Server. This report contains prompts on time and on the minimum and maximum duration of the sessions to be analyzed.

Usage scenario

Administrators can use this report to analyze the total number of sessions on the MicroStrategy system on any given day. They can also see the average,
minimum, and maximum number of sessions open at any given minute
during a day.

**Report details**

- Drill path: The recommended drill path is along the Time hierarchy.
- Other options: To restrict the scope of analysis to a specific MicroStrategy
  client such as Desktop, you can add an additional filter or page-by on the
  Connection Source attribute using the appropriate client. Alternatively,
  you can add a filter or page-by on any attribute from the Session
  hierarchy (except the Session attribute itself).

**Data load**

This analysis area covers the historical data load. There is only one report in
this area, Data Load History. This report is covered in detail below.

**Sample report: Data Load History**

This report provides a historical view of the data load process for each
project being monitored. For each project, the start and end time of each
data load is listed.

**Usage scenario**

Administrators can use this report to confirm that the statistics data is being
loaded at the proper time, and to ensure that the data load does not put an
undue load on the server.

**Report details**

- The projects are listed by project GUID.
- The project with GUID 00000000000000000 represents fact migration
  from the statistics tables of the listed project.
Delivery processing analysis

This analysis area provides reports that analyze the Distribution Services subscription activity in your system.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>111. Weekly Subscription Activity</td>
<td>Provides a comprehensive weekly summary of subscription activity.</td>
</tr>
<tr>
<td>112. Subscription Statistics</td>
<td>Provides subscription activity over a given time interval. This report is prompted on time.</td>
</tr>
<tr>
<td>113. User Subscription Statistics</td>
<td>Provides information for users on subscription execution over a given time interval. This report is prompted on time.</td>
</tr>
<tr>
<td>114. Top 10 Subscribed Documents</td>
<td>Provide insight into the documents that contribute to the top 10 percent of: recipients; execution time; or number of subscriptions over a time interval. This report is prompted on time and on which subscription analysis metric to use.</td>
</tr>
<tr>
<td>115. Top 10 Subscribed Reports</td>
<td>Provide insight into the reports that contribute to the top 10 percent of: recipients; execution time; or number of subscriptions over a time interval. This report is prompted on time and on which subscription analysis metric to use.</td>
</tr>
<tr>
<td>116. Top 10 Subscribed Contacts</td>
<td>Provide insight into the users that contribute to the top 10 percent of: recipients; execution time; or number of subscriptions over a time interval. This report is prompted on time and on which subscription analysis metric to use.</td>
</tr>
<tr>
<td>117. Top 10 Longest Executing Subscriptions</td>
<td>Lists the top 10 percent of subscriptions that contribute to the execution times over a given time period. This report is prompted on time.</td>
</tr>
</tbody>
</table>

Inbox Message Analysis

This analysis area provides reports that analyze Inbox Message activity, including errors encountered while performing actions on Inbox Messages and the contribution of Inbox Message Jobs to the total number of Jobs on Intelligence Server.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>118. Inbox Message Action over Time</td>
<td>This report provides a look at the Inbox Message Population and the number of actions performed on the set of Messages and how many of those actions resulted in errors. This report is prompted on time.</td>
</tr>
<tr>
<td>119. Number of Messages by User over Time</td>
<td>This report provides a look at the user population using the Inbox Messages. This reports looks at number of messages for each user along with the number of actions taken on these messages. This report is prompted on time.</td>
</tr>
</tbody>
</table>
Report processing analysis

A significant area of system analysis and monitoring involves tuning the server’s governing and project configuration settings. Administrators can use the reports in this analysis area to determine:

- Whether the time out setting for user sessions is appropriate. Analysis can help you configure the User Session Idle Time setting. (From Desktop, right-click a project source and select Configure MicroStrategy Intelligence Server, expand Governing, and select General.)

- Whether caching should be enabled for prompted reports. Analysis can help you configure the Enable Prompted Report Caching setting. (In Desktop, from the Folder List, expand Administration, right-click the project name, select Project Configuration, expand Caching, and select Reports (advanced)).

While insights into such questions usually involve gathering data from multiple reports spanning multiple analysis areas, the Enterprise Manager reports within the Report Processing Analysis area provide a targeted examination to assess server and project governing.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>120. Inbox Contributions to Job Counts</td>
<td>This report looks at the contribution of Inbox Message Jobs to the total Job count on Intelligence Server. This report is prompted on time.</td>
</tr>
<tr>
<td>121. Top 10 Users of Inbox Messages by Project</td>
<td>This report provides the Top 10 Users consuming the Inbox Message population by Project. This report is prompted on time.</td>
</tr>
<tr>
<td>122. Top 10 Reports in Inbox Messages by Project</td>
<td>This report provides a look at the top reports in the Inbox Message population. This report is prompted on time.</td>
</tr>
<tr>
<td>123. Top 10 Documents in Inbox Messages by Project</td>
<td>This report provides a look at the top documents in the Inbox Message population. This report is prompted on time.</td>
</tr>
<tr>
<td>124. Unread Inbox Messages</td>
<td>This report looks at the activity of Inbox Messages that have not been read. This report is prompted on time.</td>
</tr>
<tr>
<td>125. Inactive Inbox Messages</td>
<td>This report looks at the Inbox Messages that have not had any actions performed in a given amount of days. This report is prompted on a measuring metric and a number of days of inactivity.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Weekly Summary - Activity Analysis</td>
<td>Provides a comprehensive weekly summary of project activity. This report is prompted on the projects to be summarized.</td>
</tr>
<tr>
<td>2. Report Execution Analysis Working Set</td>
<td>Analyses report execution by time each job takes to execute. This report is prompted on time and on the projects to be analyzed.</td>
</tr>
</tbody>
</table>
Sample report: Database Result Rows by Report

This Enterprise Manager report can help you understand the effect on load and performance of those user reports that did not result in cache hits. This report contains a prompt on Time.

Usage scenarios

- You can use this report to identify those user reports that have high Average Elapsed Time (Average Elapsed Duration per Job) and are requested frequently (RP Number of Jobs). You can then consider a
strategy to ensure that these user reports have a high cache hit ratio in the future.

- Total Database Result Rows (RP Number of Database Result Rows) provides a good approximate measure of the size of report caches. This can give you insight into tuning report-related project settings. To make changes to the project settings, in Desktop, from the Folder List, expand Administration, right-click the project name, select Project Configuration, expand Caching, and select Reports (general). For detailed information about these settings, click Help.

- Total Database Result Rows also provides a measure of the data returned by the database to Intelligence Server for post-processing.

- Average Execution Time provides a measure of time taken to execute a report on the warehouse data source.

**Report details**

- Additional options: To restrict your analysis to a given computer, a connection source, a user session, and so on, add any attribute from the Session folder to this Enterprise Manager report. For example, to restrict analysis to Web reports, add the Connection Source attribute to the page-by axis. For detailed information about page-by, see the MicroStrategy Basic Reporting Guide.

**Resource utilization analysis (including top consumers)**

This analysis area provides reports to help you analyze how available resources are being used so you can determine optimization strategies. Available system hardware resources include various Intelligence Server machines, database servers, Web servers, and client machines. Enterprise Manager provides insight into Intelligence Server machine usage and the nature of client-side activity.

Administrators can use the reports in this analysis area to measure:

- How much time users spend in report execution queues.
- Load times to determine at what time periods peak loads occur.
- Which interfaces (Desktop, Web, Mobile, and so on) users prefer.
- Web usage statistics.
Reports in this analysis area prompt you to select a time period to analyze, and use various metrics representing answers to administrator requirements.

The Top Consumers folder contains shortcuts to reports elsewhere in Enterprise Manager. These reports, taken together, indicate what users and reports are top consumers of system resources.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>30. Execution cycle breakdown</td>
<td>Provides a daily breakdown of the time taken by each of the four steps in the report execution cycle: queue, SQL generation, SQL execution, and Analytical Engine. This report is prompted on time.</td>
</tr>
<tr>
<td>30.1 Queue time breakdown</td>
<td>Analyzes each report by the amount of time it spends in each part of the queue. This report is prompted on time.</td>
</tr>
<tr>
<td>30.2 Queue to Exec time ratios by Server Processing Unit</td>
<td>Provides a breakdown of queue time and execution time for each report job step. This report is prompted on time.</td>
</tr>
<tr>
<td>30.3 Effect of job prioritization on queue time</td>
<td>Provides an analysis of the effect of job prioritization on the queue time and execution time of reports. This report is prompted on time.</td>
</tr>
<tr>
<td>31. Activity Analysis by Weekday/Hour Working Set</td>
<td>Analyzes Intelligence Server activity on an hourly basis, based on one of a number of metrics. This report is prompted on time and on a method of analysis.</td>
</tr>
<tr>
<td>32. Peak Time Periods</td>
<td>Provides the number of jobs and the average queue and execution durations per job, on an hourly basis. This report is prompted on time.</td>
</tr>
<tr>
<td>33. Server Activity Analysis Summary</td>
<td>Analyzes the daily usage of each Intelligence Server, based on a number of metrics. This report is prompted on time and on methods of analysis.</td>
</tr>
<tr>
<td>33.1 Scheduled Report Load on Intelligence Server</td>
<td>Analyzes the duration and CPU usage of all scheduled jobs. This report is prompted on time.</td>
</tr>
<tr>
<td>33.2 Subscribed Report Load on Intelligence Server</td>
<td>Analyzes the duration and CPU usage of all NarrowCast Server subscription jobs. This report is prompted on time.</td>
</tr>
<tr>
<td>33.3.1 Web Access Trends</td>
<td>Analyzes the number of jobs run from MicroStrategy Web. This report is prompted on time.</td>
</tr>
<tr>
<td>33.3.2 Web and Non-Web Usage</td>
<td>Compares the server usage of Web and non-Web users. This report is prompted on time.</td>
</tr>
<tr>
<td>33.3.3 Web Usage Statistics</td>
<td>Provides the number of Web users, the average number of jobs per Web user, and the average report execution time per job for Web users. This report is prompted on time.</td>
</tr>
<tr>
<td>34. OLAP Cube Usage Statistics</td>
<td>Provides the overall usage of an OLAP Cube in memory. This report is prompted on time.</td>
</tr>
</tbody>
</table>
Sample report: Scheduled Report Load on Intelligence Server

This report provides a comprehensive analysis of the impact scheduled jobs have on the Intelligence Server machines in your system. This report contains a prompt on time.

Usage scenario

You can use this report to understand the daily impact of scheduled reports on each Intelligence Server machine. Impacts can be measured with metrics such as the total server report jobs and the total time spent in Intelligence Server.

You can also use this report to study which user reports are executed as part of a schedule. By viewing which scheduled jobs have errors, you can quickly take appropriate action.

Report details

• This report contains several attributes in the Report Objects window that are not in the report grid. If you have a license to use MicroStrategy OLAP
Services, you can move these attributes from the Report Objects window to the report grid without re-executing the report. For detailed information about OLAP Services, see the MicroStrategy Basic Reporting Guide.

- To know which users in your system have scheduled the largest number of jobs, include the User attribute in this report.

- To understand which schedules have been mapped to a given report in a project, include the Report attribute in this report.

- To find out which of your scheduled reports had errors, include the Error Indicator attribute in this report.

Performance analysis

Administrators can use this analysis area to understand the impact of server and project governing settings and usage patterns on the system.

The Performance Analysis folder contains a number of reports. It also contains two folders, each with its own related area of analysis:

- Cube advisor, page 695
- Performance monitoring analysis, page 696

The Performance Analysis folder in Enterprise Manager contains reports that measure such metrics as average job execution time and other job performance trends, cache analysis, longest executing reports, and so on.

Two reports from this analysis area are presented in detail below. These sample reports have been selected as representative reports of the analysis area; the details and options suggested for the sample reports can often be used on other reports within the same or related analysis areas.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>40. System Performance Trends</td>
<td>Analyzes system performance over time based on your choice of metrics. This report is prompted on time and on methods of analysis.</td>
</tr>
<tr>
<td>40.1 Average Execution Time vs. Number of Jobs per User</td>
<td>Provides a daily comparison of the number of jobs per user and the average execution time of the jobs. This report is prompted on time.</td>
</tr>
<tr>
<td>40.2 Average Execution Time vs. Number of Sessions</td>
<td>Provides a daily comparison of the number of reporting sessions and the average execution time per job. This report is prompted on time.</td>
</tr>
<tr>
<td>41. Cache Analysis</td>
<td>Analyzes the effectiveness of caching on the system. This report is prompted on time and on type of report.</td>
</tr>
</tbody>
</table>
Sample report: Cache Analysis

This report provides a comprehensive analysis of report caching within the system. A good caching strategy can significantly improve system performance. This report contains prompts on Time and on the job type you want to analyze, and allows you to select the number of top report jobs for which you want to see data.

Usage scenario

You can use this report to analyze the cache hit ratios for certain reports; typically, these are the most frequently requested or most resource-intensive reports. You can also determine whether prompted reports should be set up to create a cache by analyzing whether or not prompted reports are hitting the cache regularly.

Report details

- To analyze the cache hit ratios for element load jobs, select **Element browsing job** at the prompt. Be sure to remove the Report attribute from the report, since Element browsing jobs are ad hoc and do not map to any existing report in the metadata. This can give you insight into tuning element-related project settings. To make changes to the project settings, in Desktop, from the Folder List, expand **Administration**, right-click the project name, select **Project Configuration**, expand **Caching**, and select **Elements**. For detailed information about these settings, click **Help**.

- To analyze the cache hit ratios for prompted jobs, select **Prompted jobs** at the prompt. This can give you insight into tuning advanced
report-related project settings. To make changes to the project settings, in Desktop, from the Folder List, expand Administration, right-click the project name, select Project Configuration, expand Caching, and select Reports (advanced). For detailed information about these settings, click Help.

Sample report: Warehouse Tables Accessed

This report provides a count of the number of warehouse tables and columns accessed in various SQL clauses. This report contains a prompt on Time.

Usage scenario

You can use this report to gain insights into database tuning by determining which warehouse tables and columns are accessed in the various SQL clauses, such as SELECT, WHERE, and so on. This information can help you determine where database tuning can be adjusted to improve overall query and reporting performance of your MicroStrategy project.

For example, columns that are frequently accessed in the WHERE clause are good candidates for indexing.

Cube advisor

The reports in this analysis area give an overview of database execution statistics for report requests. They provide a picture of how measures such as report execution times are affected if reports are run against Intelligent Cubes rather than the data warehouse.

In addition, the Report Request Statistics per Project report can be used in conjunction with Cube Advisor to provide additional data on the performance benefits of potential Intelligent Cubes, such as any decrease in the time required to execute SQL statements and the number of users that would be affected by recommended Intelligent Cubes. For more information
on using this report as input for Cube Advisor, see the *Dynamic Sourcing* chapter in the *OLAP Services Guide*.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>45. DB Execution of Prompted Reports by Project</td>
<td>Provides statistics for prompted report jobs and what percentage of time is spent on the database. This report is prompted on time.</td>
</tr>
<tr>
<td>46. Report Object Listings</td>
<td>Provides analysis of the attribute and metric associations with a particular report. This report is prompted on report, or on report processing metrics.</td>
</tr>
<tr>
<td>47. Projected Cube Execution Times of Prompted Reports by Project</td>
<td>Projects the execution times of report jobs if they were to hit an Intelligent Cube. This report is prompted on time and on report processing metrics.</td>
</tr>
<tr>
<td>48. Report Request Statistics per Project</td>
<td>Provides a look at database execution statistics for report requests. This report is prompted on time and on project. <em>Note:</em> This report can be exported and used as input for the Cube Advisor.</td>
</tr>
</tbody>
</table>

**Performance monitoring analysis**

The reports in this analysis area measure Intelligence Server performance, as recorded by the Diagnostics and Performance Monitoring Tool’s Performance Monitors. For information about enabling these counters, or other uses of the Diagnostics and Performance Monitoring Tool, see *Finding trouble spots using diagnostics, page 740*.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>92. Performance Monitor Report</td>
<td>Provides a listing of Performance Monitor Counters recorded in project statistics, and their values, over a given time interval. This report is prompted on time and on what counters are populated in the Enterprise Manager Warehouse.</td>
</tr>
</tbody>
</table>
| 93. Performance vs. Governing Settings           | Provides general performance analysis against Intelligence Server governing settings over a time interval. This report is prompted on time. *Note:* To achieve maximum flexibility in this report, make sure that the following Performance Counters are logged in the Diagnostics and Performance Monitoring Tool:  
  • Executing Reports  
  • Memory Used by Report Caches  
  • Number of Report Caches  
  • Open Project Sessions  
  • Open Sessions |
Project analysis

Enterprise Manager reports within this analysis area use the Project attribute to analyze various metrics related to project use and Intelligence Server use. Administrators can use these reports to analyze project usage trends and understand how a project grows over time. The reports can help you determine which metadata objects are used and how often, so you can take appropriate actions.

The Project Analysis folder in Enterprise Manager contains the following analysis areas, each with its own reports:

- **Object properties analysis, page 698**
- **Object usage analysis, page 699**
- **Project development trends, page 704**
- **Prompt usage analysis, page 704**

These areas are described below, and one report is presented in detail. This sample report has been selected as a representative report of the analysis area; the details and options suggested for the sample report can often be used on other reports within the same or related analysis areas.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
</table>
| 94. Report Performance against Governing | Provides an analysis of report performance against Intelligence Server governing settings over a time interval. This report is prompted on time.  
**Note:** To achieve maximum flexibility in this report, make sure that the following Performance Counters are logged in the Diagnostics and Performance Monitoring Tool:  
• Executing Reports  
• Memory Used by Report Caches  
• Number of Report Caches |
| 95. User Performance against Governing | Provides an analysis of performance per user against Intelligence Server governing settings over a time interval. This report is prompted on time.  
**Note:** To achieve maximum flexibility in this report, make sure that the following Performance Counters are logged in the Diagnostics and Performance Monitoring Tool:  
• Open Project Sessions  
• Open Sessions |
| 96. Delivery Performance against Governing | Provides an analysis of performance per subscription delivery against Intelligence Server governing settings over a time interval. This report is prompted on time. |
## Object properties analysis

These reports list the properties of all objects of a specific type in the projects or Intelligence Servers monitored by Enterprise Manager.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.1 Attribute Form Properties_deprecated</td>
<td><strong>Note:</strong> This report is deprecated. Lists the properties of all attribute forms in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>50.2 Attribute Properties</td>
<td>Lists the properties of all attributes in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>50.3 Column Properties</td>
<td>Lists the properties of all columns in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>50.4 Fact Properties</td>
<td>Lists the properties of all facts in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>50.5 Hierarchy Properties</td>
<td>Lists the properties of all hierarchies in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>50.6 Logical Table Properties</td>
<td>Lists the properties of all tables in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>50.7 Transformation Properties</td>
<td>Lists the properties of all transformations in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>51.1 Consolidation Properties</td>
<td>Lists the properties of all consolidations in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>51.2 Custom Group Properties</td>
<td>Lists the properties of all custom groups in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>51.3 Document Properties</td>
<td>Lists the properties of all documents in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>51.4 Filter Properties</td>
<td>Lists the properties of all filters in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>51.5 Metric Properties</td>
<td>Lists the properties of all metrics in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>51.6 Prompt Properties</td>
<td>Lists the properties of all prompts in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>51.7 Report Properties</td>
<td>Lists the properties of all reports in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>51.8 Template Properties</td>
<td>Lists the properties of all templates in all monitored projects. This report is paged by project.</td>
</tr>
<tr>
<td>51.9 Security Filter Properties</td>
<td>Lists the properties of all security filters in all monitored projects. This report is paged by project.</td>
</tr>
</tbody>
</table>
Object usage analysis

This analysis area provides reports to analyze how objects are used within MicroStrategy.

The Object Usage Analysis folder contains a number of reports. It also contains two folders, each with its own related area of analysis:

- **Command Manager Reporting, page 701**
- **Configuration Object Usage Analysis, page 702**
- **Static Report Content Analysis, page 702**
The reports in the top-level Object Usage Analysis folder can be useful to help you determine what application objects are not being used, and can be safely deleted.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>60.1 Report Statistics</td>
<td>Lists all reports that have not been executed since the specified date, and provides the number of times they have been executed. This report is prompted on time.</td>
</tr>
<tr>
<td>60.2 Template Statistics</td>
<td>Lists all templates that have not been used since the specified date, and provides the number of times they have been used. This report is prompted on time.</td>
</tr>
<tr>
<td>61.1 Schedule Statistics</td>
<td>Lists all schedules that have not been used since the specified date, and provides the number of times they have been used. This report is prompted on time.</td>
</tr>
<tr>
<td>61.2 Server Definition Statistics</td>
<td>Lists all server definitions that have not been used since the specified date, and provides the number of times they have been used. This report is prompted on time.</td>
</tr>
<tr>
<td>62. Top 10 Reports</td>
<td>Analyzes the server load for the ten most-executed reports. This report is prompted on time.</td>
</tr>
<tr>
<td>63. Report Drilling Analysis</td>
<td>Provides information about how many times a report has been executed and how many times users have drilled from that report. This report is prompted on time.</td>
</tr>
<tr>
<td>64.1 Schedule_Report_User_Relations</td>
<td>Lists the users and reports associated with each schedule.</td>
</tr>
<tr>
<td>64.2 Schedule_Document_User_Relations</td>
<td>Lists the users and documents associated with each schedule.</td>
</tr>
<tr>
<td>65. Report Drilling Patterns</td>
<td>For a particular report, lists the objects that have been drilled from and drilled to from four-tier clients such as MicroStrategy Web. This report is prompted on time.</td>
</tr>
<tr>
<td>66. Find Patterns in Attribute-Metric Associations to Build Cubes</td>
<td>For all jobs that execute for longer than the specified time, lists attributes and metrics that are found in the same job. These attributes and metrics can then be included in an Intelligent Cube to reduce database usage. This report is prompted on Seconds.</td>
</tr>
</tbody>
</table>
## Command Manager Reporting

The reports in this analysis area provide Command Manager syntax that can be used to delete unused objects in a project, including attributes, filters, metrics, and users.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>126.1 Delete Unused Attributes</td>
<td>Provides Command Manager syntax to delete attributes that have not been associated with reports in a project. This report is prompted on the report creation date.</td>
</tr>
<tr>
<td>126.2 Delete Unused DB Instances</td>
<td>Provides Command Manager syntax to delete database instances that have no report executions associated. This report is prompted on the database instance's creation date.</td>
</tr>
<tr>
<td>126.3 Delete Unused Documents</td>
<td>Provides Command Manager syntax to delete documents that have never been executed. This report is prompted on the document's creation date.</td>
</tr>
<tr>
<td>126.4 Delete Unused Filters</td>
<td>Provides Command Manager syntax to delete filters that have not been associated with reports in a project. This report is prompted on the filter's creation date.</td>
</tr>
<tr>
<td>126.5 Delete Unused Metrics</td>
<td>Provides Command Manager syntax to delete metrics that have not been associated with reports in a project. This report is prompted on the metric's creation date.</td>
</tr>
<tr>
<td>126.6 Delete Unused Reports</td>
<td>Provides Command Manager syntax to delete reports that have never been executed. This report is prompted on the report's creation date.</td>
</tr>
<tr>
<td>126.7 Delete Unused Schedules</td>
<td>Provides Command Manager syntax to delete schedules that have never been associated with reports, documents, or users. This report is prompted on the schedule's creation date.</td>
</tr>
<tr>
<td>126.8 Delete Unused Security Filters</td>
<td>Provides Command Manager syntax to delete security filters that have no associations with users. This report is prompted on the security filter's creation date.</td>
</tr>
<tr>
<td>126.9 Delete Unused Templates</td>
<td>Provides Command Manager syntax to delete templates that are not associated with any reports. This report is prompted on the template's creation date.</td>
</tr>
<tr>
<td>126.10 Delete Unused User Groups</td>
<td>Provides Command Manager syntax to delete user groups that do not contain any users. This report is prompted on the user group's creation date.</td>
</tr>
<tr>
<td>126.11 Delete Unused Users</td>
<td>Provides Command Manager syntax to delete users that have never connected to the system. This report is prompted on the user's creation date.</td>
</tr>
<tr>
<td>126.11.1 Disable Unused Users</td>
<td>Provides Command Manager syntax to disable but not delete users that have never connected to the system. These users can be re-enabled at a later time. This report is prompted on the user's creation date.</td>
</tr>
</tbody>
</table>
Configuration Object Usage Analysis

The reports in this analysis area can be useful to help you determine what configuration objects are not being used, and can be safely deleted.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>67. Unused DB Instances</td>
<td>Lists all database instances that have not executed a job over a given time interval. This report is prompted on time.</td>
</tr>
<tr>
<td>68. Unused Projects</td>
<td>Displays all projects that have not been accessed in a given time interval, along with usage statistics for all other times. This report is prompted on time.</td>
</tr>
<tr>
<td>69. Unused Schedules</td>
<td>Displays all schedules that are not used in any reports or documents.</td>
</tr>
<tr>
<td>70. Unused Server Definitions</td>
<td>Displays all server definitions that are not loaded on Intelligence Servers, and those that are loaded but are idle, meaning users are not running jobs or connecting.</td>
</tr>
<tr>
<td>71. Unused User Groups</td>
<td>Lists all user groups that do not contain any users.</td>
</tr>
</tbody>
</table>

Static Report Content Analysis

The reports in this analysis area inform you as to what report objects are only rarely used.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>60.3 Attributes Least Used in Executed Reports</td>
<td>Lists the attributes that are used in the fewest number of jobs, and the most recent time each attribute was used. This report is filtered on time and on the number of attributes excluded.</td>
</tr>
<tr>
<td>60.4 Metrics Least Used in Executed Reports</td>
<td>Lists the metrics that are used in the fewest number of jobs, and the most recent time each metric was used. This report is filtered on time and on the number of metrics excluded.</td>
</tr>
<tr>
<td>60.5 Filters Least Used in Executed Reports</td>
<td>Lists the filters that are used in the fewest number of jobs, and the most recent time each filter was used. This report is filtered on time and on the number of filters excluded.</td>
</tr>
<tr>
<td>62.1 Top n Attributes By Usage In Executed Reports</td>
<td>Lists the most-used attributes. This report is filtered on time and on the number of attributes.</td>
</tr>
<tr>
<td>62.2 Top n Metrics By Usage In Executed Reports</td>
<td>Lists the most-used metrics. This report is filtered on time and on the number of metrics.</td>
</tr>
<tr>
<td>62.3 Top n Filters By Usage In Executed Reports</td>
<td>Lists the most-used filters. This report is filtered on time and on the number of filters.</td>
</tr>
</tbody>
</table>
Sample report: Report Drilling Patterns

This report lists the objects in each report that have been drilled from and drilled to in four-tier clients such as MicroStrategy Web. The report is paged by project and by report. It prompts you for the dates to be analyzed.

Analysis of drilling and statistics is only available from a four-tier client such as MicroStrategy Web.

Usage scenario

The Report Drilling Patterns report shows you what users want to see, by displaying the most commonly drilled-to objects. This information allows you to determine which attributes to include in a report’s list of report objects. Because SQL is not generated for OLAP Services drilling, you can use this report to optimize your OLAP Services implementation.

Sample report display

<table>
<thead>
<tr>
<th>Drill_From_Obj</th>
<th>Drill_To_Obj</th>
<th>Metrics</th>
<th>RP Number of Jobs</th>
<th>RP Number of Jobs with Error</th>
<th>RP Elapsed Duration hh:mm:ss</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUARTER</td>
<td>MONTH</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0:00:00</td>
</tr>
<tr>
<td>REGION</td>
<td>COUNTRY</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0:00:01</td>
</tr>
<tr>
<td></td>
<td>REGIONAL REVENUE BY CATEGORY</td>
<td></td>
<td>4</td>
<td>0</td>
<td>0:00:02</td>
</tr>
</tbody>
</table>

Report details

- Additional options: Use this report in conjunction with other “statistics”-type reports, which display similar usage information about individual objects such as templates, schedules, and so on.
Project development trends

The reports in this folder provide information about the objects and object usage trends in the project.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>70. Summary of Application Objects by Project</td>
<td>Provides a count of all types of application objects (reports, filters, metrics, and so on) in all monitored projects, by owner. This report is paged by project.</td>
</tr>
<tr>
<td>71. Summary of Configuration Objects by Project</td>
<td>Provides a count of all types of configuration objects (schedules, users, and so on) in all monitored Intelligence Servers, by owner. This report is paged by Intelligence Server.</td>
</tr>
<tr>
<td>72. Summary of Schema Objects by Project</td>
<td>Provides a count of all types of schema objects (facts, attributes, and so on) in all monitored projects, by owner. This report is paged by project.</td>
</tr>
<tr>
<td>73. Weekly Project Usage Trend</td>
<td>Shows the weekly trends per project of users, sessions, and requests. This report is prompted on time.</td>
</tr>
<tr>
<td>74. Weekly New Application Objects Trend Over a Selected Period</td>
<td>Provides a graph of new application objects created over a specified period. This report is prompted on time.</td>
</tr>
</tbody>
</table>

Prompt usage analysis

The reports in this folder provide information about the prompt usage and prompt answer trends in the project.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
</table>
| 1. All Prompt Answers for a Given Time Period | Lists all answers to all prompts over the specified time period. This report is prompted on time and report.  
**Note:** This report is also an as-is view of the IS_PR_ANS_FACT table for the specified time period. |
| 1.1 Most Frequently Selected Prompt Answers Per Report, Prompt Combination | Lists prompt answers for each prompt in each report, sorted by frequency. This information can help choose default prompt answers for each report. This report is prompted on time and report, and paged by prompt. |
| 1.2 Which Prompt Answer Values Are Used/Not Used For A Specific Prompt Title? | Lists the number and percentage of jobs that contain each prompt answer. This report is prompted on time, report, and prompt. |
| 1.3 Which Prompts Remain Unanswered? | Lists all optional prompts that are not answered. These prompts may be able to be safely removed from the reports. This report is prompted on time and report. |
Real-time analysis

Several administrative questions require near real-time information about project and server activity. For example:

- When a user contacts the administrator to troubleshoot an error received when executing a report, the administrator needs a list of recent errors and error messages to investigate the problem.

- Administrators often want to ensure that current throughput and response times observed by users are meeting expectations.

- Schedules are typically used to update caches during a batch window. The administrator may want to monitor the system to ensure that scheduled jobs have finished successfully.

Such requirements as those listed above focus on a relatively small snapshot of recent activity on the system. Reports that provide answers to such questions must be refreshed without requiring frequent updates using the Enterprise Manager data loader.

The Real-time Analysis reports provide details of current Intelligence Server activity. The data used in these reports is no more than 24 hours old. If a successful data load has finished in the last 24 hours, data from that data load is used; otherwise, the reports work directly with data from the statistics tables.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.1 Distribution of Prompts Within Executed Report Jobs by Prompt Type, Location Type</td>
<td>Provides information about the prompts and prompt locations in executed reports. This report is prompted on time and report.</td>
</tr>
<tr>
<td>1.5 Prompt Answers That Result in Reporting Errors</td>
<td>Provides a list of prompt answers that cause errors in report execution, and the corresponding errors. This report is prompted on time and report.</td>
</tr>
</tbody>
</table>
The reports in this analysis area use Freeform SQL and provide targeted administrative reporting features that complement the historical reporting features in the Operations, Performance, Project, and User Analysis areas.

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>101. Recently Completed Jobs</td>
<td>Provides details about all jobs that have completed since the specified date. This report is prompted on time.</td>
</tr>
<tr>
<td>102. Recent Sessions, Users</td>
<td>Provides details about recent user connection activity This report is prompted on time.</td>
</tr>
<tr>
<td>103. Recently Completed Scheduled Jobs</td>
<td>Provides details about all recently completed scheduled jobs. This report is prompted on time.</td>
</tr>
</tbody>
</table>

**User analysis**

Reports in this analysis area contain the User attribute as their primary attribute for analysis, along with various metrics that answer an administrator’s questions about user activity and preferences.

The User Analysis folder in Enterprise Manager contains the following analysis areas, each with its own reports:

- *User activity analysis, page 706*
- *User preference analysis, page 709*

These areas are described below, and two reports are presented in detail. These sample reports have been selected as representative reports of the analysis area; the details and options suggested for the sample reports can often be used on other reports within the same or related analysis areas.

**User activity analysis**

This analysis area provides reports to analyze the effects of user activity on the system. Administrators can measure the following:

- Who are the most prolific users in terms of number of jobs, connection duration, and so on?
- How are users using features such as ad-hoc reporting, drilling, and so on?
- Which users are using the system correctly and which need more training, as identified by the number of cancelled jobs and jobs with errors?

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>80. Top N users</td>
<td>Determine the top N users, based on a metric of your choice. This report is prompted on time and on analysis method.</td>
</tr>
<tr>
<td>81. Activity by User</td>
<td>Provides summary information of user reporting activity by user and project. This report is prompted on time.</td>
</tr>
<tr>
<td>81.1 Ad-hoc job activity by User</td>
<td>Provides information about how many ad-hoc jobs are being run, and the composition of ad-hoc jobs. This report is prompted on time.</td>
</tr>
<tr>
<td>81.1.1 Drilling Activity by User</td>
<td>Provides information about how many jobs each user has run, and how many of those jobs were drilling. This report is prompted on time.</td>
</tr>
<tr>
<td>81.2 DB Result Rows by User</td>
<td>Provides the number of jobs, the number of database result rows and the average elapsed report execution duration per user and project. This report is prompted on time.</td>
</tr>
<tr>
<td>82. User Statistics</td>
<td>Lists all users that have not logged in since the specified date, and provides information about their connections. This report is prompted on time.</td>
</tr>
<tr>
<td>83. User Inactivity by Project</td>
<td>Provides a list of users that have not connected to a project over a certain time interval. It also provides the last connection and disconnection timestamp.</td>
</tr>
</tbody>
</table>

**Sample report: Activity by User**

This report provides data on total elapsed report duration. It also provides counts of cancelled jobs, non-cancelled jobs, and jobs that end with an error, as well as timed-out jobs by user and by project. This report prompts the user for a time for the analysis.

**Usage scenario**

You can use this report to gain insight into how reports are used on a per project basis by all users. You can determine which users are wasting resources by repeatedly cancelling jobs, and determine which users run the most reports in a given project. You can also see where reporting errors originate.
**Report details**

- **Drill paths:**
  - To narrow the scope of your analysis to individual sessions, drill across from User to Session and keep the parent attribute.
  - To identify the reports and documents that were executed by a given user during a session, drill across from Session to Report/Document.

- **Other options:**
  - To restrict your analysis to the most prolific users based on your chosen criteria, add the report Top N Users as a filter to this Enterprise Manager report.
  - To determine which projects a given user is using, add a filter on User.
  - To restrict your analysis to a given machine or connection source, add any attribute from the Session folder to this Enterprise Manager report.

**Sample report: Top N Users**

This report displays the top N users based on the user activities you select. The report prompts you for user activities and the number of users you want returned.

**Usage scenario**

You can use this report to learn the top users in a number of analysis areas related to user activity, including:

- Which users log in to Intelligence Server most often. (Select the Number of Sessions metric.)
- Which users are connected for the longest duration. (Select the Connection Duration metric.)
- Which users run the most report jobs. (Select the RP Number of Jobs metric.)

**Report details**

- Add your own metrics to this report for user activity analysis that focuses on your environment’s requirements.
• Use this report as a filter in custom reports that you create. For example, the Activity by User report returns the total elapsed time for report execution by user and project, as well as the number of cancelled and non-cancelled jobs. You can add this Top N Users report as a filter to the Activity by User report, to narrow the results to the top ten users responsible for the highest number of cancelled jobs. This allows you to analyze overall user activity and determine whether these users are cancelling jobs legitimately.

**User preference analysis**

This analysis area provides reports to analyze the effects of user group activity on the system. Administrators can measure the following:

• What groups do specific users belong to?
• What reports are most commonly executed by members of a specific user group?

<table>
<thead>
<tr>
<th>Report name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>90. List User Groups to which users belong</td>
<td>Lists all User Groups to which the specified users belong. This report is prompted on user.</td>
</tr>
<tr>
<td>91. Popular reports in a user’s User Group</td>
<td>Lists the top N most-executed reports in a user’s User Group. This report is prompted on user, time, and number of reports.</td>
</tr>
</tbody>
</table>
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, it 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 Health Center Configuration Wizard.

This chapter contains the following information:

- *Accessing the Health Center console, page 712*
- *Configuring and managing a Health Center system, page 712*
- *Diagnosing and fixing problems, page 718*
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 Start menu, point to Programs, then MicroStrategy, then Tools, 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. By default, the installation folder is C:\Program Files\Common Files\MicroStrategy\Health Center. If a folder whose name matches this machine's name is not found, Health Center does not start. 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 mstrsuppapp.

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 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 714*.

- Configure all machines that are running MicroStrategy server products, such as Intelligence Server, MicroStrategy Web Server, 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 Desktop 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 to the Health Center system on a case-by-case basis for troubleshooting.

- When creating a Health Center system, first connect to the Master Health Agent, and then add the Health Agents, either manually or by discovering Health Agents in your network. Only one Master Health Agent is required for each Health Center system, and the Master Health Agent must be the first machine added to the system.

- 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.

• 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 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 722.

• 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 Health Center Configuration Wizard to configure a machine as a Master Health Agent. For information about each page of the wizard, click Help.

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 will not start.

To start the Health Center Configuration Wizard

- In Windows, from the Start menu, point to Programs, then MicroStrategy, then Tools, then select Health Center Configuration Wizard.
- In UNIX/Linux, in the Health Center installation directory, in the /bin folder, run the program mstrsuppappcfgwiz.

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 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 impact on performance.

Configuring a machine as a Health Agent

Use the Health Center Configuration Wizard to configure a machine as a Health Agent. For information about each page of the wizard, click Help.

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 will not start.
To start the Health Center Configuration Wizard

- In Windows, from the Start menu, point to Programs, then MicroStrategy, then Tools, then select Health Center Configuration Wizard.

- In UNIX/Linux, in the Health Center installation directory, in the /bin folder, run the program mstrsuppappcfgwiz.

Adding Health Agents to the system

When you add a Health Agent to the Health Center system, the Master Health Agent can then perform tasks such as scheduling system checks or viewing log files for that machine.

You can add 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. Both 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 715.

If a Health Agent machine is behind a firewall, the Master Health Agent may not be able to connect to it. In this case, you need to configure the Master Health Agent to wait for an incoming connection from the Health Agent. For instructions, see Adding a Health Agent from behind a firewall, page 717.

To add a 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.

   - 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 then 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, the Master Health Agent may not be able to connect directly to the Health Agent. In this case, you must configure the Master Health Agent to receive a connection from the Health Agent.

### To add a Health Agent that is behind a firewall to the network

1. When you configure the machine as a Health Agent, on the Health Agent Connectivity page of the Health Center Configuration Wizard, select the **Connect automatically to** check box, and then type the IP address and port number of the Master Health Agent to connect to.

2. 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.
3 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.

The **Health Agent name** field is case-sensitive.

4 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 723*.

---

**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 723*.

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 only be scheduled to run on a daily basis, and some can only be scheduled to run on a weekly basis. The 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 on a daily basis. For details on each system check, see List of system checks, page 723.

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 on a weekly basis. For details on each system check, see List of system checks, page 723.

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.

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).
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 720*.

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** subdirectory.

6 To transmit the diagnostics package to MicroStrategy Technical Support via FTP or FTPS:
   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.

### List of system checks

A system check is a Health Center scan of a specific 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 725
  - Logging Settings, page 725
- System checks for Desktop, page 725
  - Component Registration, page 725
  - LogPath Registry Key, page 725
  - Metadata Connection, page 726
• **System checks for Enterprise Manager, page 726**
  - Data Load Failures, page 726

• **System checks for Intelligence Server, page 727**
  - Component Registration, page 727
  - Directories Available Space, page 727
  - Directories Permissions, page 728
  - Logging Settings, page 728
  - LogPath Registry Key, page 728
  - Memory Depletion, page 729
  - Memory Governing, page 729
  - Metadata Connection, page 730
  - Network Port Detector, page 730
  - Path Environment Variable, page 731
  - Warehouse Connections, page 731

• **System checks for Narrowcast Server, page 731**
  - Component Registration, page 731
  - DCOM Settings, page 732
  - Logging Settings, page 732
  - Network Ports, page 732
  - Object Repository Connection, page 733
  - Path Environment Variable, page 733

• **System checks for MicroStrategy Web products, page 733**
  - Directories Permissions, page 733
  - JVM Maximum Heap Size, page 734
  - JVM Value, page 734
  - Temp Folder Available Space, page 734
System checks for Configuration

Logging Settings

This system check allows you to change what information is logged from various MicroStrategy products, such as Desktop, 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 Desktop

Component Registration

This system check verifies that all .EXE, .DLL, and OCX files required by Desktop are present and properly registered on this machine.

This system check is only 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

This system check confirms that the LogPath registry key exists and is pointing to a valid location for Desktop log files.

If the registry key points to an invalid location, you can modify the log path by clicking Modify MicroStrategy Desktop Log Path. Type the log path in the field and click OK.
**Metadata Connection**

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, you are instructed to launch the MicroStrategy Configuration Wizard.

**System checks for Enterprise Manager**

**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 whether 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 amount of time (in hours) that this system check will allow a data load to proceed before reporting an error. The default value is 5 hours.

If the data load service is present 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 only 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.

Directories Available Space

This system check verifies whether 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 a different storage location with more available space should be chosen.

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 of the directories are 1024 MB and 10% of total disk space.
Directories Permissions

This system check verifies whether 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 all of these directories, this system check fails. To resolve this problem, either 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

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:

- 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

This system check analyzes Intelligence Server for potential memory depletion problems. It performs the following actions:

• Checks to see whether Intelligence Server's Memory Contract Manager is enabled and properly configured

• Checks to see whether the operating system has allocated enough RAM swap space for Intelligence Server to perform adequately

• On Windows systems, checks to see whether 4GT mode is enabled

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.

• Only 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 360.
- 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.
- On Windows machines, if 4GT mode is not enabled, click the link provided to enable it.

**Metadata Connection**

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, you are instructed to launch the MicroStrategy Configuration Wizard.

**Network Port Detector**

This system check analyzes the Intelligence Server machine to determine whether the network ports used by MicroStrategy products 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 Test 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 cannot be scheduled.
Path Environment Variable

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 only 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

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.

If a warehouse connection is not present, you are instructed 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 only 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.
**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 whether 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.

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 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, you are instructed to launch the Narrowcast Server Administrator.

Path Environment Variable

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 only 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

Directories Permissions

This system check verifies whether 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 all of these directories, this system check fails. To resolve this problem, either 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 does have read and write permissions.
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 Server 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 Windows, you can choose to restart IIS when you change the JVM maximum heap size.

### JVM Value

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 whether the machine has enough physical memory to support the JVM Max Heap Size as specified in the registry.

If the JVM Max Heap Size is greater than the amount of physical memory available on this machine, you can type a new value for the maximum heap size in the **JVM Max Heap Size** field.

On Windows, you can choose to restart IIS when you change the JVM maximum heap size.

### Temp Folder Available Space

This system check verifies whether enough disk space is available for MicroStrategy Web to function properly. It checks the available disk space for the operating system’s Temp directory.
If this directory 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, either some files should be deleted from the machine hosting the directory, or a different storage location with more available space should be chosen.

On the **Configuration** tab, specify the minimum free disk space that must be available for the Temp directory, 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 are 1024 MB and 10% of total disk space.
Introduction

This chapter provides guidance for finding and fixing trouble spots in the system. While the material in 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 738*
- *Finding trouble spots using diagnostics, page 740*
- *Memory depletion troubleshooting, page 759*
- *Authentication troubleshooting, page 765*
- *Fixing inconsistencies in the metadata, page 773*
- *Object dependencies troubleshooting, page 776*
- *Date/time functions troubleshooting, page 776*
- *Performance troubleshooting, page 777*
- *Troubleshooting data imported from a file, page 779*
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 740*).
- Use the server state dump (the DSSErrors log file) (see *Analyzing a server state dump, page 755*) to determine whether it was a:
  - Memory depletion (see *Memory depletion troubleshooting, page 759*)
  - Exception error (see *Working with exceptions, page 755*)
- Tune the system as necessary (see *Chapter 7, Tuning your System for Best Performance*)
**The system is slow**

- Use reports in MicroStrategy Enterprise Manager (see *Chapter 14, Analyzing System Usage with Enterprise Manager*) 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 7, Tuning your System for Best Performance*).

**Users cannot log in**

- See *Authentication troubleshooting, page 765*

Intelligence Server will not start

- The connection to metadata may not be working (see *Connecting to the MicroStrategy metadata, page 8*)

Intelligence Server will 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 9*)
- 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 780*)

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 806*)
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 360) 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 740
• Viewing and analyzing log files, page 752
• Analyzing a server state dump, page 755

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 within 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 Desktop: 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 **Desktop Preferences**. In the **General** category, in the **Advanced** subcategory, select the **Show Diagnostics Menu Option** check box and click **OK**.
   
     - In Windows: From the **Start** menu, point to **Programs**, then **MicroStrategy**, then **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.

```
<table>
<thead>
<tr>
<th>Component</th>
<th>Dispatcher</th>
<th>System Log</th>
<th>Console Log</th>
<th>File Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Default)</td>
<td>Error</td>
<td></td>
<td>✔</td>
<td>DSErrors</td>
</tr>
<tr>
<td></td>
<td>Fatal</td>
<td></td>
<td>✔</td>
<td>DSErrors</td>
</tr>
<tr>
<td></td>
<td>Info</td>
<td></td>
<td>✔</td>
<td>DSErrors</td>
</tr>
<tr>
<td></td>
<td>Warning</td>
<td></td>
<td>✔</td>
<td>DSErrors</td>
</tr>
<tr>
<td></td>
<td>Scope Trace</td>
<td></td>
<td></td>
<td>&lt;None&gt;</td>
</tr>
<tr>
<td>Analytical Engine</td>
<td>Error</td>
<td></td>
<td>✔</td>
<td>DSErrors</td>
</tr>
<tr>
<td></td>
<td>Scope Trace</td>
<td></td>
<td></td>
<td>&lt;None&gt;</td>
</tr>
<tr>
<td></td>
<td>Major Trace</td>
<td></td>
<td></td>
<td>&lt;None&gt;</td>
</tr>
<tr>
<td></td>
<td>Scope Trace</td>
<td></td>
<td></td>
<td>&lt;None&gt;</td>
</tr>
<tr>
<td>Authentication Server</td>
<td>Error</td>
<td></td>
<td>✔</td>
<td>DSErrors</td>
</tr>
<tr>
<td></td>
<td>Memory Trace</td>
<td></td>
<td></td>
<td>&lt;None&gt;</td>
</tr>
<tr>
<td></td>
<td>Scope Trace</td>
<td></td>
<td></td>
<td>&lt;None&gt;</td>
</tr>
<tr>
<td></td>
<td>Trace</td>
<td></td>
<td></td>
<td>&lt;None&gt;</td>
</tr>
<tr>
<td>Behavior</td>
<td>Errors</td>
<td></td>
<td></td>
<td>&lt;None&gt;</td>
</tr>
<tr>
<td></td>
<td>Scope Trace</td>
<td></td>
<td></td>
<td>&lt;None&gt;</td>
</tr>
<tr>
<td>Calculation Server</td>
<td>Error</td>
<td></td>
<td>✔</td>
<td>DSErrors</td>
</tr>
</tbody>
</table>
```
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 only for this machine, select Machine Default.
- To configure logging for the server instance, select CastorServer Instance.

To configure the server instance with the logging settings that are currently used by this machine, select CastorServer Instance and then select the Use Machine Default Configuration check box.

**Configure diagnostics logging**

3 Select the Diagnostics Configuration tab. For more information about diagnostics logging, see Diagnostics configuration, page 743.

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 751.

**Configure performance logging**

7 Select the Performance Configuration tab. For more information about performance logging, see Configure performance logging settings, page 745.

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 Viewing and analyzing log files, page 752.

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:

- **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
particular component and its related trace (dispatcher). To add a combination, select its check box.

<table>
<thead>
<tr>
<th>Component</th>
<th>Dispatcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Server</td>
<td>Trace</td>
</tr>
<tr>
<td>Database Classes</td>
<td>All</td>
</tr>
</tbody>
</table>
| 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 323.*
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 751.

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.

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.

<table>
<thead>
<tr>
<th>MicroStrategy Feature</th>
<th>Components</th>
<th>Trace Level</th>
</tr>
</thead>
</table>
| 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  
• DSS ClientConnection                                           | • Authentication 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. |

© 2011 MicroStrategy, Inc.
<table>
<thead>
<tr>
<th>MicroStrategy Feature</th>
<th>Components</th>
<th>Trace Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Group Editor</td>
<td>• DSS CommonDialogsLib&lt;br&gt;• DSS CommonEditorControlsLib&lt;br&gt;• DSS Components&lt;br&gt;• DSS DateLib&lt;br&gt;• DSS EditorContainer&lt;br&gt;• DSS EditorManager&lt;br&gt;• DSS EditorSupportLib&lt;br&gt;• DSS ExpressionboxLib&lt;br&gt;• DSS FilterLib&lt;br&gt;• DSS FTRContainerLib&lt;br&gt;• DSS ObjectsSelectorLib&lt;br&gt;• DSS PromptEditorsLib&lt;br&gt;• DSS PromptsLib</td>
<td>All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.</td>
</tr>
<tr>
<td>Data Transmitters and Transformers</td>
<td>• DSS DataTransmitter&lt;br&gt;• DSS MhtTransformer&lt;br&gt;• DSS MIME&lt;br&gt;• DSS SMTPSender&lt;br&gt;• DSS Network</td>
<td>• Function Level Tracing</td>
</tr>
<tr>
<td>Element Browsing</td>
<td>• DSS DBElementServer&lt;br&gt;• DSS ElementNetClient&lt;br&gt;• DSS ElementNetServer&lt;br&gt;• DSS ElementServer</td>
<td>All components perform Element Source Tracing. DSS DBElementServer also performs Report Source Tracing.</td>
</tr>
<tr>
<td>Fact Creation Wizard</td>
<td>• DSS FactCreationWizard&lt;br&gt;• DSS ProgressIndicator</td>
<td>Function Level Tracing</td>
</tr>
<tr>
<td>Fact Editor</td>
<td>• DSS ColumnEditor&lt;br&gt;• DSS CommonDialogsLib&lt;br&gt;• DSS Components&lt;br&gt;• DSS EditorContainer&lt;br&gt;• DSS EditorManager&lt;br&gt;• DSS ExpressionboxLib&lt;br&gt;• DSS ExtensionEditor&lt;br&gt;• DSS FactEditor</td>
<td>All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.</td>
</tr>
<tr>
<td>Filter Editor</td>
<td>• DSS CommonDialogsLib&lt;br&gt;• DSS CommonEditorControlsLib&lt;br&gt;• DSS Components&lt;br&gt;• DSS DateLib&lt;br&gt;• DSS EditorContainer&lt;br&gt;• DSS EditorManager&lt;br&gt;• DSS EditorSupportLib&lt;br&gt;• DSS ExpressionboxLib&lt;br&gt;• DSS FilterLib&lt;br&gt;• DSS FTRContainerLib&lt;br&gt;• DSS ObjectsSelectorLib&lt;br&gt;• DSS PromptEditorsLib&lt;br&gt;• DSS PromptsLib</td>
<td>All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.</td>
</tr>
<tr>
<td>MicroStrategy Feature</td>
<td>Components</td>
<td>Trace Level</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Hierarchy Editor</td>
<td>• DSS CommonDialogsLib, DSS EditorContainer, DSS EditorManager, DSS HierarchyEditor</td>
<td>Function Level Tracing</td>
</tr>
<tr>
<td>HTML Document Editor</td>
<td>• DSS CommonDialogsLib, DSS Components, DSS DocumentEditor, DSS EditorContainer, DSS EditorManager</td>
<td>All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.</td>
</tr>
<tr>
<td>Metadata SQL</td>
<td>• DSS MD4Server</td>
<td>• Object Tracing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Access Tracing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SQL Tracing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Content Source Tracing</td>
</tr>
<tr>
<td>Metric Editor</td>
<td>• DSS CommonDialogsLib, DSS Components, DSS DimtyEditorLib, DSS EditorContainer, DSS EditorManager, DSS ExpressionboxLib, DSS MeasureEditorLib, DSS PromptsLib, DSS PromptsLib, DSS PropertiesControlsLib</td>
<td>All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.</td>
</tr>
<tr>
<td>Object Browsing</td>
<td>• DSS ObjectServer, DSS SourceNetClient, DSS SourceNetServer</td>
<td>All components perform Content Source Tracing. DSS ObjectServer also performs Object Source Tracing.</td>
</tr>
<tr>
<td>Partition Editor</td>
<td>• DSS CommonDialogsLib, DSS Components, DSS DataSliceEditor, DSS EditorContainer, DSS EditorManager, DSS FilterLib, DSS PartitionEditor</td>
<td>All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.</td>
</tr>
<tr>
<td>Print Schema</td>
<td>• DSS PrintCore, DSS PrintSchema, DSS ProgressIndicator</td>
<td>Function Level Tracing</td>
</tr>
<tr>
<td>Project Creation</td>
<td>• DSS AttributeCreationWizard, DSS FactCreationWizard, DSS ProgressIndicator, DSS ProjectCreationLib, DSS WHCatalog</td>
<td>Function Level Tracing</td>
</tr>
<tr>
<td>Project Duplication</td>
<td>• DSS AsynchLib, DSS ProgressIndicator, DSS ProjectUpgradeLib, DSS SchemaManipulation</td>
<td>Function Level Tracing</td>
</tr>
<tr>
<td>MicroStrategy Feature</td>
<td>Components</td>
<td>Trace Level</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Project Upgrade</td>
<td>• DSS AsynchLib</td>
<td>Function Level Tracing</td>
</tr>
<tr>
<td></td>
<td>• DSS ProgressIndicator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS ProjectUpgradeLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS SchemaManipulation</td>
<td></td>
</tr>
<tr>
<td>Prompt Editor</td>
<td>• DSS CommonDialogsLib</td>
<td>All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.</td>
</tr>
<tr>
<td></td>
<td>• DSS CommonEditorControlsLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS Components</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS EditorContainer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS EditorManager</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS EditorSupportLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS PromptEditorsLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS PromptStyles</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS SearchEditorLib</td>
<td></td>
</tr>
<tr>
<td>Report Editor</td>
<td>• DSS CommonDialogsLib</td>
<td>All components perform Function Level Tracing. DSS Components also performs Explorer and Component Tracing.</td>
</tr>
<tr>
<td></td>
<td>• DSS CommonEditorControlsLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS Components</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS DateLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS EditorContainer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS EditorManager</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS EditorSupportLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS ExportLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS ExpressionBoxLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS FilterLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS FTRContainerLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS GraphLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS GridLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS ObjectsSelectorLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS PageByLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS PrintGraphInterface</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS PrintGridInterface</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS PromptEditorsLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS PromptsLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS PropertySheetLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS RepDrillingLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS RepFormatsLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS RepFormsLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS ReportControl</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS ReportDataOptionsLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS ReportSortsLib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DSS ReportSubtotalLib</td>
<td></td>
</tr>
<tr>
<td>Report Execution</td>
<td>• DSS ReportNetClient</td>
<td>• Report Source Tracing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Process Tracing</td>
</tr>
<tr>
<td></td>
<td>• DSS ReportNetServer</td>
<td>• Process Tracing</td>
</tr>
<tr>
<td></td>
<td>• DSS ReportServer</td>
<td>• Report Source Tracing</td>
</tr>
<tr>
<td>MicroStrategy Feature</td>
<td>Components</td>
<td>Trace Level</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 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                                                     |
| 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. |
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`, while 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.

---

**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\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 753.

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 so that only the relevant entries are shown, such as only viewing entries relating to memory conditions for times when you are trying to troubleshoot a memory depletion. For more information about viewing log files in Health Center, see Chapter 15, Maintaining your MicroStrategy System with Health Center.

The MicroStrategy Web server error log files are located 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

<table>
<thead>
<tr>
<th>Section</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Numeric ID of the process that performed the action</td>
</tr>
<tr>
<td>thread</td>
<td>Numeric ID of the thread that performed the action</td>
</tr>
<tr>
<td>date::time</td>
<td>Date and time at which the action happened</td>
</tr>
<tr>
<td>module name</td>
<td>Name of the MicroStrategy component that performed the action</td>
</tr>
<tr>
<td>trace type</td>
<td>Type of the log file entry</td>
</tr>
<tr>
<td>message</td>
<td>Message about the action</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Log Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intelligence Server creates a report definition.</td>
</tr>
<tr>
<td>2</td>
<td>Intelligence Server loads the report definition object named Length of Employment from the metadata.</td>
</tr>
<tr>
<td>286:[THR:480][02/07/2003::12:24:23:860][DSS ReportServer][Report Source Tracing] where Definition = Object(Name=&quot;Length of Employment&quot; Type=3(Report Definition) ID=D1AE564911D5C4D04C200E8820504F4F Proj=B19DEDCC11D4E0EFC000EB9495D0F44F Ver=493C8E3447909F1FBF75C48E11AB7DEB)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Intelligence Server creates a report instance named Length of Employment.</td>
</tr>
<tr>
<td>286:[THR:480][02/07/2003::12:24:24:931][DSS ReportServer][Report Source Tracing]Created ReportInstance(Name=&quot;Length of Employment&quot; ExecFlags=0x1000180(OSrcCch UptOSrcCch) ExecActn=0x1000180(RslvCB LclCch))</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Intelligence Server begins executing the report instance.</td>
</tr>
<tr>
<td>286:[THR:480][02/07/2003::12:24:24:931][DSS ReportServer][Report Source Tracing]Executing ReportInstance(Name=&quot;Length of Employment&quot; ExecFlags=0x1000180(OSrcCch UptOSrcCch) ExecActn=0x1000180(RslvCB LclCch)) with Actions=0x8300003f(Rslv GenSQL ExeSQL Alrt XTab EvalVw LclCch UptLclCch), Flags=0x1000180(OSrcCch UptOSrcCch)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Intelligence Server checks to see whether the report exists in the report cache.</td>
</tr>
<tr>
<td>6</td>
<td>Intelligence Server did not find the report in the cache.</td>
</tr>
<tr>
<td>286:[THR:480][02/07/2003::12:24:25:342][DSS ReportServer][Report Source Tracing]Not found in cache: ReportInstance(Name=&quot;Length of Employment&quot; ExecFlags=0x1000180(OSrcCch UptOSrcCch) ExecActn=0x1000180(RslvCB LclCch))</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Intelligence Server checks for prompts and finds none in the report.</td>
</tr>
<tr>
<td>286:[THR:314][02/07/2003::12:24:25:432][DSS ReportServer][Report Source Tracing]No prompts in ReportInstance(Name=&quot;Length of Employment&quot; ExecFlags=0x1000180(OSrcCch UptOSrcCch) ExecActn=0x1000180(RslvCB LclCch))</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Intelligence Server executes the report and updates the caches.</td>
</tr>
<tr>
<td>286:[THR:492][02/07/2003::12:24:26:634][DSS ReportServer][Report Source Tracing]Executing ReportInstance(Job=2 Name=&quot;Length of Employment&quot; ExecFlags=0x1000184(OSrcCch UptOSrcCch) ExecActn=0x1000184(ExeSQL RslvCB LclCch)) with Actions=0x3000003f(Rslv GenSQL ExeSQL Alrt XTab EvalVw LclCch UptLclCch), Flags=0x1000184(OSrcCch UptOSrcCch)</td>
<td></td>
</tr>
</tbody>
</table>
More detail is logged for report execution if the report is run from MicroStrategy Desktop.

**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 current 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 within 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 759) or exceptions (see Working with exceptions, page 755). 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, currently 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 pre-amble

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 that 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 Desktop, right-click the project source, and select Configure MicroStrategy Intelligence Server). The settings include:

- 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 currently not listed in a 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
• History List settings
• Idle timeouts
• XML governors
• Memory Contract Manager (MCM) settings

MCM is specifically designed to help you avoid memory depletions. For more information on MCM, see Governing Intelligence Server memory use with Memory Contract Manager, page 360.

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 11, Scheduling Jobs and Administrative Tasks.

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 8.

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 current 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 within 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 within 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 will enable your system to be stable and reliable. Intelligence Server’s memory use is discussed earlier in this guide (see Memory, page 355). 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 or not certain job tasks are allowed to occur based on how much memory they could consume. This does not guarantee that a memory depletion will not happen, but decreases the chance of it. For more information on this, see Governing Intelligence Server memory use with Memory Contract Manager, page 360.

- Be proactive. During the project planning/building phase, apply those concepts that are presented in the Tuning section. For information, see Chapter 7, 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 353) and MicroStrategy Enterprise Manager (see Chapter 14, Analyzing System Usage with Enterprise Manager). 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 7, 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 355.

**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 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.

**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?

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:

- Use the Windows Performance Monitor (see *Monitoring memory use with Performance Monitor, page 357*) 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 *Viewing and analyzing log files, page 752*)

- Your knowledge of the system and whether or not 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 762*
- *Report cache memory, page 762*
- *Working set, page 763*
- *History List (Inbox) filled with messages, page 763*
- *A single large report, page 764*
- *Export to Excel from MicroStrategy Web or Web Universal, page 765*
- *Multiple memory-intensive requests on Linux, page 765*
Intelligence Server memory footprint

When Intelligence Server starts, a number of operations occur, all of which use memory. For detailed information about this, see the following sections:

- *What happens when Intelligence Server starts?, page 15*
- *How much memory does Intelligence Server use when it starts up?, page 358*
- *How does Intelligence Server use memory after it is running?, page 359*

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 the amount of time they can exist (see *Managing database connection threads, page 386* )
- Reduce the Maximum RAM size for report caches (see *Maximum RAM usage, page 483*, 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 456*. 
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 483*)

**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 374*)
- Reduce the working set size
- Reduce size of reports
- Limit the user sessions either in number or by time (*Governing active users, page 372*)

**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 active users, page 372*)
- Limit the number of Inbox Maximum number of messages per user (see *Governing user resources, page 374*).
A single large report

You may notice that when a certain report is run, memory use spikes. This could be caused by a number of 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 of the solutions below, see Results processing, page 394 (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 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 508). Just 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 399*)
  - Reduce the Maximum number of XML cells setting
  - Reduce the Maximum number of results 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 Web?, page 52*.

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 399*)

**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. Refer to the appropriate authentication section below.
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:

1. On the MicroStrategy Web server machine(s), access the IIS Internet Service Manager. The Microsoft Management Console dialog box opens.

2. Right-click the MicroStrategy virtual folder and select Properties.

   - 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 MicroStrategy Desktop in MicroStrategy Intelligence Server Universal, you might see the following error message: “MsiSessionManager: failed 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, re-create the server definition as follows:

1. Launch the MicroStrategy Configuration Wizard. (In Windows, from the Start menu point to Programs, then MicroStrategy, 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 Desktop via a 3-tier project source.

Troubleshooting LDAP authentication

LDAP authentication problems within Intelligence Server usually fall into one of these categories:

- Authentication issues that include clear text and Secure Socket Layer (SSL) authentication modes
- Functionality problems/questions about importing users or groups, and synchronizing LDAP users within 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 located in the appropriate directory. See *Implementing LDAP authentication, page 126* for details on choosing the correct DLL files based on your SDK and installing them in the correct location. 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 “Can’t 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 authentication 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 authentication against LDAP: **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 MicroStrategy Desktop 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 than 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 MicroStrategy 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 MicroStrategy Intelligence Server machine within the correct system path?
  - Do the LDAP server side logs show success messages? The LDAP administrator can access these logs.

**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 server(s). Once the authentication is successful (Intelligence Server has verified the existence of the LDAP user within 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 126 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 than 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.

The authentication user DN is specified on the Intelligence Server Configuration Editor, in the LDAP: Configuration category in the User distinguished name (DN) box.

The user’s DN is specified on the User Editor, Authentication tab in the User distinguished name (DN) box.

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 within the LDAP Users folder because it is maintained within 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.

**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 only save items 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, based on the group they become part of.

May two different users have different LDAP links, but the same user name?

No. The MicroStrategy metadata may not contain two users with the same login name or user name. If you attempt to create a user with the same user login or user name, the import fails. Each user object in the MicroStrategy metadata must have a unique user login and user name.

What happens if there are two users with similar descriptions in the LDAP directory?

If the DN descriptor that specifies a particular user is not sufficient for Intelligence Server to identify him/her, then 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 will not readily apply to the user. For this to work, you will need to manually assign the user as a member of the group after she or he has been imported.
Fixing inconsistencies in the metadata

Logical inconsistencies in the metadata may arise due to 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 only to that particular object. This way you do not have to search through all the objects in a project.

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Object Classification</th>
<th>Object Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>DssTypeUnknown</td>
<td>The type of object is not specified</td>
</tr>
<tr>
<td>0</td>
<td>DssTypeReserved</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>DssTypeFilter</td>
<td>Filter</td>
</tr>
<tr>
<td>2</td>
<td>DssTypeTemplate</td>
<td>Template</td>
</tr>
<tr>
<td>3</td>
<td>DssTypeReportDefinition</td>
<td>Report</td>
</tr>
<tr>
<td>4</td>
<td>DssTypeMetric</td>
<td>Metric</td>
</tr>
<tr>
<td>5</td>
<td>Unused</td>
<td>None</td>
</tr>
<tr>
<td>6</td>
<td>DssTypeAutostyles</td>
<td>Autostyle</td>
</tr>
<tr>
<td>7</td>
<td>DssTypeAggMetric</td>
<td>Base formula</td>
</tr>
<tr>
<td>8</td>
<td>DssTypeFolder</td>
<td>Folder</td>
</tr>
<tr>
<td>9</td>
<td>Unused</td>
<td>None</td>
</tr>
<tr>
<td>10</td>
<td>DssTypePrompt</td>
<td>Prompt</td>
</tr>
<tr>
<td>11</td>
<td>DssTypeFunction</td>
<td>Function</td>
</tr>
<tr>
<td>12</td>
<td>DssTypeAttribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Object Type</td>
<td>Object Classification</td>
<td>Object Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>13</td>
<td>DssTypeFact</td>
<td>Fact</td>
</tr>
<tr>
<td>14</td>
<td>DssTypeDimension</td>
<td>Hierarchy</td>
</tr>
<tr>
<td>15</td>
<td>DssTypeTable</td>
<td>Logical table</td>
</tr>
<tr>
<td>16</td>
<td>Unused</td>
<td>None</td>
</tr>
<tr>
<td>17</td>
<td>DssTypeFactGroup</td>
<td>Fact group</td>
</tr>
<tr>
<td>18</td>
<td>DssTypeShortcut</td>
<td>Shortcut (a reference to another object)</td>
</tr>
<tr>
<td>19</td>
<td>DssTypeResolution</td>
<td>Saved prompt answer</td>
</tr>
<tr>
<td>20</td>
<td>Unused</td>
<td>None</td>
</tr>
<tr>
<td>21</td>
<td>DssTypeAttributeForm</td>
<td>Attribute form</td>
</tr>
<tr>
<td>22</td>
<td>DssTypeSchema</td>
<td>Schema (the collection of objects that define the data warehouse structure)</td>
</tr>
<tr>
<td>23</td>
<td>DssTypeFindObject</td>
<td>Search definition (a simple search)</td>
</tr>
<tr>
<td></td>
<td>Note: This object type is deprecated. Search objects are object type 39 (DssTypeSearch).</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>DssTypeCatalog</td>
<td>Catalog (a list of relevant tables in a database)</td>
</tr>
<tr>
<td>25</td>
<td>DssTypeCatalogDefn</td>
<td>Catalog definition (a description of how a catalog is constructed)</td>
</tr>
<tr>
<td>26</td>
<td>DssTypeColumn</td>
<td>Column (a property needed to define a column of a database table)</td>
</tr>
<tr>
<td>27</td>
<td>DssTypePropertyGroup</td>
<td>Property group (an internal object used to cache lists of property sets)</td>
</tr>
<tr>
<td>28</td>
<td>DssTypePropertySet</td>
<td>Property set (an internal object)</td>
</tr>
<tr>
<td>29</td>
<td>DssTypeDBRole</td>
<td>Database Instance</td>
</tr>
<tr>
<td>30</td>
<td>DssTypeDBLogin</td>
<td>Database Login</td>
</tr>
<tr>
<td>31</td>
<td>DssTypeDBConnection</td>
<td>Database Connection</td>
</tr>
<tr>
<td>32</td>
<td>DssTypeProject</td>
<td>Project</td>
</tr>
<tr>
<td>33</td>
<td>DssTypeServerDef</td>
<td>Server definition (a description of a configuration of an Intelligence Server)</td>
</tr>
<tr>
<td>34</td>
<td>DssTypeUser</td>
<td>MicroStrategy User or Group</td>
</tr>
<tr>
<td>35</td>
<td>Unused</td>
<td>None</td>
</tr>
<tr>
<td>36</td>
<td>DssTypeConfiguration</td>
<td>Intelligence Server configuration (a top level object representing a MicroStrategy installation)</td>
</tr>
<tr>
<td>37</td>
<td>DssTypeRequest</td>
<td>Scheduled request</td>
</tr>
<tr>
<td>Object Type</td>
<td>Object Classification</td>
<td>Object Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>38</td>
<td>Unused</td>
<td>None</td>
</tr>
<tr>
<td>39</td>
<td>DssTypeSearch</td>
<td>Search object</td>
</tr>
<tr>
<td>40</td>
<td>DssTypeSearchFolder</td>
<td>Search folder (a folder-like object used to store the result of a search object)</td>
</tr>
<tr>
<td>41</td>
<td>Unused</td>
<td>None</td>
</tr>
<tr>
<td>42</td>
<td>DssTypeFunctionPackageDefinition</td>
<td>Function Definition</td>
</tr>
<tr>
<td>43</td>
<td>DssTypeRole</td>
<td>Transformation</td>
</tr>
<tr>
<td>44</td>
<td>DssTypeSecurityRole</td>
<td>Security Role</td>
</tr>
<tr>
<td>45</td>
<td>DssTypeInBox</td>
<td>History folder, or inbox (a folder-like object used to store History messages)</td>
</tr>
<tr>
<td>46</td>
<td>DssTypeInBoxMsg</td>
<td>History message (an object that describes the status of a report execution request)</td>
</tr>
<tr>
<td>47</td>
<td>DssTypeConsolidation</td>
<td>Consolidation</td>
</tr>
<tr>
<td>48</td>
<td>DssTypeConsolidationElement</td>
<td>Consolidation element</td>
</tr>
<tr>
<td>49</td>
<td>DssTypeScheduleEvent</td>
<td>Schedule event (an event that can trigger a scheduled object)</td>
</tr>
<tr>
<td>50</td>
<td>DssTypeScheduleObject</td>
<td>Scheduled object (an object that can be triggered)</td>
</tr>
<tr>
<td>51</td>
<td>DssTypeScheduleTrigger</td>
<td>Scheduled trigger (the binding between schedule event and object)</td>
</tr>
<tr>
<td>52</td>
<td>DssTypeLink</td>
<td>Link (the holder of a property that spans objects)</td>
</tr>
<tr>
<td>53</td>
<td>DssTypeDBTable</td>
<td>Physical database table</td>
</tr>
<tr>
<td>54</td>
<td>DssTypeTableSource</td>
<td>Same suffix and prefix tables (a collection of physical database tables with the same suffix and prefix)</td>
</tr>
<tr>
<td>55</td>
<td>DssTypeDocumentDefinition</td>
<td>Report Services Document</td>
</tr>
<tr>
<td>56</td>
<td>DssTypeDrillMap</td>
<td>Drill map</td>
</tr>
<tr>
<td>57</td>
<td>DssTypeDBMS</td>
<td>DBMS definition (an object that holds information about a physical database)</td>
</tr>
<tr>
<td>58</td>
<td>DssTypeMDSecurityFilter</td>
<td>Security filter</td>
</tr>
</tbody>
</table>

Manual editing of values in the MicroStrategy metadata incorrectly may cause serious, project-wide problems that may make the project unusable. Since these are user-initiated changes, they are not covered by any MicroStrategy warranty. Users are strongly encouraged to back up their metadata prior to any alteration.
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 upon modules, which are activated by keys. Each module can contain a number of tests as well as 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 will address your problem. For more information on contacting Technical Support see Technical Support, page xxxvii.

Object dependencies troubleshooting

If you attempt to delete multiple objects at once in Desktop 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. While we do 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* in Chapter 1, Understanding the MicroStrategy System Architecture.

## 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, there are some things you can look at:

- **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 Desktop, right-click the project source and select **Configure MicroStrategy Intelligence Server**. In the Projects category, 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 Desktop, 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 Desktop, from the Tools menu, select Diagnostics. Click Help for details on the various tracing options.)

### 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 517](#).

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 740](#).

### 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 OLAP 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:` then the report is connecting to one or more Intelligent Cubes. The Intelligent Cubes the report is connecting to are listed below this 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.

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 Desktop, right-click a project, select **Project Configuration**, expand **Governing**, 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 due to the prompt that requires an answer. The subscription is therefore invalidated and automatically removed from Desktop’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 will not appear on the subscription’s list of contacts. If you 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 585*.)

- 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 585*.)
• 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 MicroStrategy Desktop, 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 (yyyy 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 Desktop, 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.

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:

```
```

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 MicroStrategy Desktop 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**.

In order for the change to take effect, you must restart the MicroStrategy Intelligence Server which 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 different language than project or object default language**

If the object does not contain translations for either the project default language or the object default language, then 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 will be displayed but will 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*.

Failure to start MicroStrategy Intelligence Server

Logon failure

When you attempt to start Intelligence Server, you may receive the following error message:

Failed to start service

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 Start menu, point to Program Files, then MicroStrategy, then 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 within 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.
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.

**Clustered environments troubleshooting**

**Problems in a clustered environment**

![Warning icon] 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.

**Which node am I connected to?**

In Desktop, the project source definition includes the server name. Desktop only connects to a specific node, 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 68*.)

**Do I need multiple project sources?**

Because Desktop 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 Desktop?**

Nodes in the cluster can host a different set of projects from the same metadata. The node to which a Desktop user connects is important because it dictates which projects will be available to the user at that time.

**Can I control which node I connect to in 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 will increment 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 IS_CACHE_HIT_STATS table.

How do I automate the deletion of caches on multiple nodes?

Use MicroStrategy Command Manager to run scripts that automatically delete caches. See Chapter 12, 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 will not be propagated to any Intelligence Servers connected to the metadata. Additionally, if an object is modified from an Intelligence Server not in the cluster but 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 12, 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 physically located 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 445.

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 323.

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 7, 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 792.

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 correctly located within the Enterprise Manager Warehouse. When the statistics tables are created using the MicroStrategy Configuration Wizard, they must be created in the database that will 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:
  ```sql
  SELECT * FROM IS_SESSION_STATS;
  ```

- Check report-related statistics:
  ```sql
  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:
  ```sql
  SELECT * FROM IS_SCHEDULE_STATS;
  ```

- Check cache-related statistics:
  ```sql
  SELECT * FROM IS_CACHE_HIT_STATS;
  ```

- Check document-related statistics:
  ```sql
  SELECT * FROM IS_DOCUMENT_STATS;
  SELECT * FROM IS_DOC_STEP_STATS;
  ```

For a detailed list of the statistics tables and columns, see the Enterprise Manager Data Model and Object Definitions, in the 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 (crash).

- 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 Enterprise Manager warehouse, which also hosts the statistics tables. For a list of certified databases, see the MicroStrategy Enterprise Manager readme file.

- 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 Chapter 14, Analyzing System Usage with Enterprise Manager.

- If you are using single instance session logging and the specified project is not one of the monitored projects, then no data for any monitored project is logged. To check this, open the Intelligence Server Configuration Editor to the Statistics 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 instance session logging, see Logging all statistics from a project source to the same database, page 327.

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.
• Intelligence Server may have stopped logging statistics. For possible actions you can take to correct this, see *Statistics are not being logged*, page 790.

### 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. These statistics are collected within Enterprise Manager, so you can use Enterprise Manager features to report on the data.

The majority of this information is logged in the schedule statistics table IS_MESSAGE_STATS. A detailed list of the information contained in this statistics table can be found in the *Statistics Data Dictionary* in the *Reference for System Administration*. The table below shows the pertinent Narrowcast Server information and the equivalent information in the Enterprise Manager statistics tables.

<table>
<thead>
<tr>
<th>Narrowcast Server Service Information Category</th>
<th>Distribution Services Schedule IS_MESSAGE_STATS data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>IS_MESSAGE_STATS : PHYSICALADDRESS</td>
</tr>
<tr>
<td>User ID</td>
<td>IS_MESSAGE_STATS : RECIPIENTCONTACTID and/or RECIPIENTCONTACTNAME</td>
</tr>
<tr>
<td>Login ID</td>
<td>Not applicable. All users log in as themselves and use USERID.</td>
</tr>
<tr>
<td>Unique Message ID</td>
<td>Create a unique message ID by combining IS_MESSAGE_STATS : MESSAGEINDEX with IS_MESSAGE_STATS : SESSIONID</td>
</tr>
<tr>
<td>Subscription Set ID</td>
<td>IS_MESSAGE_STATS : SCHEDULEID. The Narrowcast Server concept of a subscription set is a schedule in Distribution Services.</td>
</tr>
<tr>
<td>Address ID</td>
<td>IS_MESSAGE_STATS : ADDRESSID</td>
</tr>
<tr>
<td>Subscription ID</td>
<td>IS_MESSAGE_STATS : SUBSINSTID</td>
</tr>
</tbody>
</table>
Additional resources for troubleshooting

You can improve your troubleshooting efforts by using all of the troubleshooting best practices and information resources available. Information to access each resource is described below.

MicroStrategy readme and release notes

The 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 they include links to additional resources for further information on a topic. They are well-organized, indexed, and provide a left-hand menu listing related information based on 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 Programs, MicroStrategy, and select ReadMe.
- **UNIX/Linux**: Double-click the readme.htm file located in the install folder where you installed MicroStrategy.

Readme

The readme provides general information on the MicroStrategy release, as well as information on the different 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.

- **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, seven-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 the following URL: [https://resource.microstrategy.com/Support/](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 the following URL:

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 Desktop, 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. For more information on Technical Support and how best to ensure a timely solution to your questions, see *Technical Support, page xxxvii* in the *Book Overview and Additional Resources* chapter of this guide.
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 MicroStrategy Desktop 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, there are other administrative parameters that 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 appendix, each option in the Mobile Server or Web Server administration interface is documented in the relevant Help system. To access this Help system, click Help in Web Server Administrator or Mobile Server Administrator.

Topics covered in this appendix include:

- Assigning privileges for MicroStrategy Web, page 800
- Using the MicroStrategy Web Administrator page, page 802
- Defining project defaults, page 804
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 228.

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:

- 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 within 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 225.
Using the MicroStrategy Web Administrator page

The Administrator page allows you to connect and disconnect MicroStrategy Web or Web Universal products to MicroStrategy 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 815.

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 entitled “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 MicroStrategy 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 Assigning privileges to users and groups, page 79.

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/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.
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 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 488.

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 the 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
- 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 only displayed under certain circumstances. For example, the Report Services link only appears if you have a license to use Report Services.

Each category comprises 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.
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.

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 MicroStrategy 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 807 to learn how.

3 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 Desktop, 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 Desktop, 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.

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 Desktop Help.

### Working with SSL (HTTPS)

**SSL (secure sockets layer)** is an encryption technology that encodes the communication over the Internet so that only the recipient can read it. A URL beginning with `https://` indicates that you are communicating using SSL.

Setting up and using SSL between Web browsers and MicroStrategy Web products or between mobile devices and MicroStrategy Mobile, is a process that is totally external to MicroStrategy products. You do not need to do anything special for MicroStrategy Web products to use SSL. The following steps describe the high-level process for enabling SSL on your Web server.

#### To use SSL

1. You need to get a digital ID (also known as an authentication certificate or digital certificate) from a trusted third-party source (certificate authority), which can verify your identity. You can get certificates from companies like Verisign and Thawte or you can create your own.

2. Once you have the certificate, you need to install the certificate on your Web server and enable SSL. Refer to your Web server documentation for information on installing the certificate and configuring the Web server to use SSL.

3. The client browsers need to have a corresponding certificate that enables them to communicate with the secure Web server. The secure Web server and browser automatically exchange certificate information when establishing a connection. If the certificate can be authenticated by a trusted certificate authority, the secure page opens automatically.

   When using SSL, MicroStrategy Web and Mobile products communicate with the client on the port number designated by
your web server for SSL. The default port number for SSL may vary according to web server, but it is often 443. Check your web server’s documentation for the default port number. If you have a firewall between your clients and web servers and you are using SSL, you must make sure the appropriate port is allowed to send and receive requests through the firewall.

Most browsers come with a pre-defined list of trusted certificate authorities that include Verisign and Thawte, and sometimes GeoTrust/Equifax. If you connect to a secure server authenticated by an authority not included in your browser’s list (if you created your own, for example), or if any of the information does not match, or, in some cases, if you use an old browser that needs an update, you may get a security warning. In most cases you will be able to proceed and have an encrypted connection using SSL.

4 For MicroStrategy Web and Web Universal, once you have a secure Web server, all you need to do is use https:// to ensure secure communication between your Web clients and the Web server. For example, the URL to access MicroStrategy Web in your environment is http://machine_name/microstrategy/asp, where machine_name is the name of the Web server. If you enable SSL, you would use the following URL instead: https://machine_name/microstrategy/asp

5 For MicroStrategy Mobile, when you configure your mobile devices, you must make sure they are configured to use a Request Type of https instead of http. For information about configuring mobile devices, see the Administration chapter of the MicroStrategy Mobile Design and Administration Guide.

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, it uses 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 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**

There is a setting in the Security section of the Administrator Page that allows you to encrypt all communication between the Web server and Intelligence Server. There is quite a bit of additional overhead involved in encrypting and decrypting all this network traffic so you may see a noticeable performance degradation if you choose to do this. However, if you are working with particularly 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.
3. Select the **Traffic between the Web and Iserver machines is encrypted** check box.
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 select few 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\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:

1. External firewall:
   - HTTP or HTTPS
   - MicroStrategy Web user login and password encrypted if using SSL

2. Internal firewall:
   - TCP/IP communication to MicroStrategy Intelligence Server (port 34952 by default)
   - Application Security based on MicroStrategy user login with either standard or NT authentication
   - If MicroStrategy user login is mapped to warehouse user and password, it is encrypted in the metadata
   - Metadata password encrypted in NT registry
   - Data Warehouse Database
   - Metadata Database

MicroStrategy user login and password encrypted when transferred over the network to Intelligence Server.

TCP/IP communication over the network to the MicroStrategy Intelligence Server (port 34952 by default).
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\n
   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

   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 chapter Deploying MicroStrategy
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.

How do time-out settings in MicroStrategy Web and Intelligence Server affect MicroStrategy Web users?

There are three settings related to session time-out that may affect MicroStrategy Web users.

First, in the Intelligence Server Configuration Editor, under Governing - General, the Web user session idle time 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\MicroStrategy\Web ASPx, the time-out setting determines the length of time (in 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:

```xml
<sessionState
    mode="InProc"
    stateConnectionString="tcpip=127.0.0.1:42424"
    sqlConnectionString="data source=127.0.0.1;user id=sa;password="
```
This setting does not affect Web Universal, since it does not use .NET architecture.

The third setting is the MicroStrategy Web Administration setting **Allow automatic login if session is lost**. This setting can be found 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 these settings interact in various combinations.

<table>
<thead>
<tr>
<th>Intelligence Server time-out</th>
<th>web.config time-out</th>
<th>Allow automatic login if session is lost</th>
<th>User idle time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>45 minutes</td>
<td>20 minutes</td>
<td>Either</td>
<td>30 minutes</td>
<td>User must log back in</td>
</tr>
<tr>
<td>20 minutes</td>
<td>45 minutes</td>
<td>No</td>
<td>30 minutes</td>
<td>User must log back in</td>
</tr>
<tr>
<td>20 minutes</td>
<td>45 minutes</td>
<td>Yes</td>
<td>30 minutes</td>
<td>User is automatically logged back in</td>
</tr>
<tr>
<td>20 minutes</td>
<td>45 minutes</td>
<td>Either</td>
<td>60 minutes</td>
<td>User must log back in</td>
</tr>
</tbody>
</table>

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

- Clustering multiple web servers together improves performance. For more information about this, see *Chapter 8, 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.
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 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.

application object A MicroStrategy object used to provide analysis of and insight into relevant data. Application objects are developed in MicroStrategy Desktop 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.

**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).

**child dependency** Occurs when an object uses other objects in its definition.

See also [parent dependency](#).

**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 users** Users who execute reports or use the system in one way or another in the same time.

**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.

**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 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 MicroStrategy Desktop 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 MicroStrategy Intelligence Server uses to connect to a particular database.

See also:
- login ID
- password

**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 MicroStrategy Desktop machines so they can be retrieved more quickly.

encryption  The translation of data into a sort of secret code for security purposes.

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.

**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 OLAP Services add-on for Intelligence Server.

job  A request to the system, created by users submitting requests from Desktop, 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.

**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 MicroStrategy Desktop or Web.

**object cache** A recently used object definition stored in memory on MicroStrategy Desktop and MicroStrategy Intelligence Server.

**ODBC** See Open Database Connectivity.

**ODBC driver** A software routine that translates MicroStrategy 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.

**parent dependency** Occurs when an object is used as part of the definition of other objects.

See also **child dependency**.

**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 MicroStrategy Desktop 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 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 a MicroStrategy 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 MicroStrategy 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.

**scheduling**  
A MicroStrategy 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 MicroStrategy Desktop. 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 MicroStrategy 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.

**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.

**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 (2^{32}). 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 (Web, Desktop, Narrowcast Server, and so on).

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 MicroStrategy 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.
INDEX

Numerics
6.X database authentication mode 126

A
access control
   application functionality 69
to data via the database 109
Access Control List defined on 71
   assigning 70
   controlling application access 76
Object Manager 273
User Merge Wizard 114
   viewing 66
accessing
   Contact Editor 587
   Device Editor 579
   Performance Monitor 353
   Transmitter Editor 573
ACL. See Access Control List.
active user defined on 371
addresses for contacts 592
administration tasks, scheduling 560
Administrator page in MicroStrategy Web 802
   accessing 802
   alert 565
   analyzing statistics 669
   anonymous authentication mode 122
   application development life cycle 240
      developer 243
      recommended 241
   application object defined on 5
      copying 258
   applying project defaults for MicroStrategy Web 804
   architecture overview 1
   archiving History List messages 498
   assigning
      permissions 70
      privileges individually 80
      privileges with security role 82
   AT command. See Windows AT command.
auditing for license compliance 228
authentication defined on 117
   example 217
   implementing Windows 169
   LDAP 126
   LDAP binding 160
   LDAP password comparison 160

© 2011 MicroStrategy, Inc.
standard 120
   troubleshooting 765
authentication mode 117
   6.X database 126
   anonymous 122
automating a process with Command Manager 595

B
baseline-versus-baseline integrity test 613
baseline-versus-project integrity test 613
best practices
   cache management 459
   contacts 587

C
   cache defined on 455
      best practices 459
      centralized cache file 430
      database connection 10
      element. See element cache.
      in a cluster 452
      object. See object cache.
      purging for a project 30
      purging for a server 31
      result cache types 469
      sharing 430
      types 455
      viewing 467
   Cache Monitor 466
   callstack 757
   canceling a job 56
   centralized cache file 430
   certificate, digital 809
   change journaling 30, defined on 315
      comments 317
   enabling 315
   exporting entries 321
   filtering entries 319
   logged information 315
   project 315
   project source 316
   purging 322
   searching entries 319
   viewing entries 318
   changing project status 35
   child dependency defined on 263
      locating 263
      Object Manager 267
   cluster defined on 422
      See also clustering.
   Cluster Monitor 30
      accessing 31
      using 31
   Cluster view
      accessing 31
      System Maintenance Monitor 30
      using 31
   clustering defined on 421
      architecture 425
      backing up caches 477
      benefits 422
      caches 452
      components 422
      Intelligence Server machines 441
      monitoring 30
      requirements 422
      sharing a History List across a cluster 431
      sharing information between nodes 426
      sharing metadata across a cluster 427
      Web 422
      combining users 113
command line
   Integrity Manager 628
   Integrity Manager exit codes 630
Command Manager 595
   encrypting scripts 604
   execution errors 606
   privileges 596
   running 597, 601
   scheduling with Windows AT command 596
   syntax errors 605
   timeout errors 607
comma-separated value file for export 53
comparing reports 611
   baseline-versus-baseline comparison 613
   baseline-versus-project comparison 613
   project-versus-project comparison 613
   viewing differences 632
complying with license 228
concurrent user defined on 371
configuration object defined on 6
   copying 258
   moving 258
configuring
   Health Agent 715
   JAAS 192
   Master Health Agent 715
   system architecture 347
conflict resolution default action 272
connection borrowing defined on 391
connection mapping defined on 88
   example 218
   purpose 88
   User Merge Wizard 115
Contact Editor, accessing 587
contact group 589
contacts
   about 566
   address maintenance 592
   best practices 587
   Contact Editor 587
   copying 588
   creating 585
   deleting 588, 592
   disabling 588
   duplicating 588
   enabling 588
   example 567, 585
   linked user maintenance 591
   prerequisites 586
   recommended tasks 586
   renaming 588
   searching for dependent subscriptions 588
cookie defined on 810
copying
   application object 258
   configuration object 258
   object 253
   project 241, 246
   resolving a conflict when 269
   schema object 258
CPU affinity 230
   for MicroStrategy Web 235
CPU license
   managing 225
   verifying 224
creating
   contact 585
   contact group 589
   device 577
   group 67
   log file 752
   schedule 547
Index

transmitter 573
user 67
creating integrity test 617
CSV, exporting as 53
Cube Monitor 524

D
data access
collection mapping 88
passthrough execution 93
security filter 95
security using RDBMS 109
security view 110
splitting a fact table by column 112
splitting a fact table by row 110
data loading
manual 662
scheduling Enterprise Manager 657
troubleshooting 664
data security 88
data warehouse defined on 4
authentication by 123
collection mapping 88
collection thread 386
DBMS upgrading 13
design and tuning 348
Enterprise Manager 643
leveraging the security view 90
limiting connection threads 762
monitoring connection to 11
monitoring usage trends 640
navigating in 4
optimizing connection threads 388
performance 347
placement 350
platform 348
setting optimal number of connection threads 386
SQL maximum size 385
SQL timeout per pass 385
timestamp 331, 661
troubleshooting 739
data warehouse connection,
governing 387
database authentication mode 123
database connection
busy 10
cache 10
deleting 11
Enterprise Manager 326
governing summary 416
information in the SSD 758
restricting element caching 514
viewing current 11
database connection cache 10
governing 389
Database Connection Monitor 11
database connection thread
governing 387
optimizing 388
prioritizing 390
timeout 389
dataset defined on 48
default location for the Web log file 753
default options for MicroStrategy
Web 804
applying 804
loading 804
user preferences 804
deleting
database connection 11
device 584
element cache 516
History List messages 501
object cache 520
report cache 468
transmitter 576
user 67
deployment, sizing 352
designing a Narrowcast Server application 418
device
  about 566
  best practices 578
  creating 577
  deleting 584
  Device Editor 579
  environment 566
  example 577
  recommended tasks 578
Device Editor 579
diagnosing system problems using Health Center 718
diagnostics defined on 740
  configuring 743
  configuring a performance counter 745
  log file anatomy 753
  sample log file 753
Diagnostics and Performance Logging tool 338, 740
diagnostics package 722
  creating in Health Center 722
  sending to technical support 723
digital certificate 809
disabling
  change journaling comments 317
  contacts 588
  document caching 466
  element caching 507
  object caching 518
  report caching 466
  single report caching 487
disconnecting
  database 11
user 68
Distinguished Name (DN) defined on 142
troubleshooting LDAP 770
Distribution Services 564
document defined on 47
  Enterprise Manager 673
  instance 48
  system performance and 406
  testing execution of 613
document cache
  file format 462
  index file 463
  managing 470
  monitoring 466
  viewing 467
document instance defined on 48
double-byte characters 608
drill path, XML 78
duplicating a project 241, 246

E

element browsing defined on 45
  job processing steps 45
element cache defined on 505
  deleting 516
  disabling 507
  enabling 507
  limiting attribute elements 513
  limiting memory usage 512
  limiting number of elements 508
  matching algorithm 507
  optimizing 510
  pool 506
  restricting by database connection 514
  restricting by database login 515
  settings 516
  storage location 506
Index

terminology 506
element incremental fetch
  optimizing 510
  setting 508
element request 506
enabling
  contacts 588
  element caching 507
  single report caching 487
encryption 604, defined on 812
  metadata 5
  with SSL 809
Enterprise Manager 639
  administering 659
  Administrator user group 646
  architecture 641
  best practices 641
  best practices for installing and configuring 646
  configuring 645, 646
  console 650
  data load, scheduling 657
  data warehouse in DB2 647
  data warehouse in Sybase 647
  database connection 326
  DB2 for the data warehouse 647
  document in 673
  EMAdmin user group 659
  Freeform SQL and 706
  installing 645, 646
  log statistics 332
  metadata database 649
  permissions 671
  project 669
  project documentation 672
  real-time analysis 705
  renaming a project 661
  statistics troubleshooting 790
  Sybase for the data warehouse 647
  system check in Health Center 726
  timestamp 661
  user 659
  Web Viewer user group 646
event triggering
  automatic 551
  manual 552
  event-based schedule 548
example
  authentication 217
  connection mapping 218
  contacts 567, 585
  fact table partitioning 218
  MCM 368
  memory depletion 368
  partitioned fact table 218
  passthrough execution 217
  project and data warehouse maintenance scenarios 37
  standard authentication 218
  Windows authentication 217
Excel file with formatting, exporting as 53
exception 755
execute permission 77
executing a job 38
  canceling 56
  current job 55
  governing 381
  governing by project 381
  governing by user 382
  prioritizing 390
  seeing the current job 55
  SSD information 758
  viewing the current job 55
executing an integrity test 618
  command line 628
  Integrity Manager Wizard 618
execution idle project mode 33
execution time, governing 383
expiring a report cache 475
export size, governing 402
exporting
  change journal entries 321
  CSV 53
  Excel file with formatting 53
  PDF 54
  plain text 53
  report from Web 52, 402

F
fact table partitioning
  example 218
fact table partitioning as a security measure 110
  example 218
failover 423
failover support defined on 423
fatal exception 755
filtering
  change journal 319
  History List messages 504
firewall defined on 806
Freeform SQL 672
deleting managed objects 309
  Enterprise Manager and 706
full idle project mode 34

G
governing
  database connection cache 389
  database connection thread 387
  execution time 383
  export size 402
  History List 374
  jobs per project 381
  jobs per user 382
  memory for Web requests 370
  number of jobs 380, 381
  report size 395
  request 379
  results delivery 399
  settings 407
  SQL 385
  user profile 377
  user resources 374
  user session 372
  working set 375
  XML cell 401
  XML drill path 404
  graph in Integrity Manager 633

H
Health Agent 715
  adding to Health Center system 716
  configuring 715
  finding (discovering) 716
  firewall and 717
Health Center 711
  accessing 712
  adding Health Agents to 716
  console 712
  diagnostic package 722
Enterprise Manager system check finding (discovering) Health Agents 716
fixing problems found by Health Agent 715
Health Agent configuration 715
Intelligence Server system check log file in 720
manual system check 718
Master Health Agent 714
Master Health Agent configuration 715
MicroStrategy Web system check 733
Narrowcast Server system check network 712
scheduled system check 719
system check 723
Health Center Configuration Wizard 715
Health Center console 712
high watermark (HWM) defined on 365
history cache defined on 460
Cache Monitor 467
History List defined on 488
accessing 496
archiving 498
backing up 496
caching and clustered environment and 496
deleting messages 501
governing 374
lifetime 501
managing 500
sharing across a cluster 431
History List message deleting 501
filtering 504
purging 504
viewing details 504
HTML document defined on 49
instance 50
job processing steps 49
HTML document instance defined on 50
HWM. See high watermark.

I
idle time defined on 374
idling a project 30, 36
IIS tuning 819
Import Data, deleting managed objects 309
importing group 67
user 67
user group 67
inbox synchronization defined on 431
inbox, SSD information 758
incremental fetch in Web 400
index file
document cache 463
report cache 462
inheriting privileges 82
installation, silent 57
Integrity Manager 611
creating an integrity test 617
graph 633
integrity test creation 617
loading saved test settings 619
prerequisites for using 612
prompted report and report data 632
report execution output 635
saving test settings 619
test results, viewing 631
viewing differences in reports 632
viewing saved results of a test 634, 637
viewing test results 631
### Integrity Manager Wizard
- Defined on: 612, 613
- Integrity test: 613, 618
- Baseline-versus-baseline: 613
- Baseline-versus-project: 613
- Creating: 617
- Executing from command line: 628
- Executing with Integrity Manager Wizard: 618
- Loading: 619
- Project-versus-project: 613
- Report status: 631
- Saving: 619
- Single-project: 613
- Viewing results of: 631
- Viewing saved results: 634, 637

### Intelligence Server
- Component, MCM: 360
- Components information in the SSD: 759
- CPU type and speed: 353
- CPUs, number of: 353
- Diagnostics: 740
- Governing: 408
- Job processing: 38
- Logging: 740
- Number of CPUs: 353
- Request types: 38
- Running mode: 16
- Runtime memory use: 359
- SSD components information: 759
- SSD version information: 756

#### Starting
- As a service: 18
- As an application: 23
- Automatically: 20
- Troubleshooting: 739

#### Startup memory use: 358

#### Stopping: 16

#### System check in Health Center: 727

### Thread balancing: 394

### Troubleshooting: 739

### Unregistering as a service: 17

### Version in SSD: 756

### Intelligent Cube
- Defined on: 523
- Cube Monitor: 524
- Job processing steps: 43
- Viewing: 524
- International support: xxxvi
- Invalidating a report cache: 468, 472

### J

#### JAAS. See Java Authentication and Authorization Service.

#### Java Authentication and Authorization Service: 192

#### Job
- Defined on: 38
- Execution: 386
- Governing: 380
- Job Monitor: 55
- Viewing currently executing job: 55

#### Job priority
- Defined on: 390
- Setting: 392
- Variables: 391

#### Job processing steps: 38
- Element browsing: 45
- Graph report: 43
- HTML document: 49
- Intelligent Cube report: 43
- Object browsing: 43
- Report: 40
- Report with a prompt: 42
- Web: 51
- Web exporting: 52

#### JVM heap size: 819
L

latency and project failover 449
LDAP defined on 126
anonymous/guest user 153
binding authentication 160
clear text 132
connection pooling 164
Connectivity Wizard 134
database passthrough 161
directory defined on 126
group search filters 138
host 132
importing
  at login 142
  in batch 143
  users and groups 141
IP address 132
linking
  users and groups 152
  Windows login 162
password comparison only 160
port 132
SDK 129
  UNIX/Linux 131
search filter
  list of groups 146
  list of users 145
search root 136
server defined on 126
SSL 132
  certificate 133
synchronization schedule 158
troubleshooting 168
user group 64
user search filter 137
LDAP authentication
FAQ 771
  troubleshooting 767
LDAP authentication. See LDAP authentication.
license
  auditing 228
  check 223
  compliance and CPU affinity 232
  CPU 224
  managing 225
  out of compliance 225
  time to run license check 223
  updating 229
  verifying named user 222
License Manager 226
Lightweight Directory Access Protocol. See LDAP.
Linux, install LDAP SDK 131
load balancing defined on 423
loaded project mode 32
loading
  project 30, 35
  project defaults for MicroStrategy Web 804
  report cache from disk 468
  saved report cache 471
  local cache file 427
locking a project 255
log file
  creating 752
  diagnostics 753
  Health Center and 720
  location 752
  reading 752
  viewing 752
logging configuration 740
logging subscriptions statistics 793
login ID, using to restrict element caching 515
low watermark (LWM) defined on 365
LWM. See low watermark.
M

machine size recommendations 352
managed object defined on 309
deleting from a project 311
managing
  license 225
  object individually 256
  project configuration 29
  project security 29
  project status 29
Master Health Agent 714
  best practices for selecting 714
  configuring 715
matching caches defined on 460
matching-history cache defined on 461
maximum single allocation size 361
maximum use of virtual address space 362
MCM
criteria for granting a request 363
  example 368
  governing Intelligence Server memory use 360
  granting or denying memory requests 362
logic flow 364
managing virtual address space fragmentation 366
maximum use of virtual address space 362, 365
memory request idle time 362, 367
memory watermark 365
minimum reserved memory 362, 365
preventing memory depletion 759
single memory allocation governing 361
tasks governed 360
using 361
XML and 361
MDX cube source, deleting managed object 309
memory 355, 760
governing for Web requests 370
SSD information 757
top consumers 761
Memory Contract Manager. See MCM.
memory depletion 759
  example 368
  preventing 759
  SSD information 757
memory request idle mode defined on 366
memory request idle time 362
  recommended value 367
merging
  groups 113
  projects 296
  users 113
message lifetime defined on 501
metadata defined on 4
corr "metadata synchronization" defined on 427
MicroStrategy Integrity Manager 611
MicroStrategy Mobile
  firewall 806
  managing 799
  security 805
  SSL (HTTPS) 809
MicroStrategy Office, installing from Web 816
MicroStrategy Project Comparison Wizard 304
MicroStrategy Project Merge 296
MicroStrategy user group. See user group.
MicroStrategy User Merge Wizard 113
MicroStrategy user. See user.
MicroStrategy Web 753
    Administrator page 802
    cookie 811
    CPU affinity feature 235
default options 804
edition 800
    encryption 812
    file-level security 813
    firewall 806
Flash Mode 800
log file 753
managing 799
Narrowcast Server and 815
security 805
SSL (HTTPS) 809
system check in Health Center 733
user settings 804
MicroStrategy Web Universal, security roles with 803
migrating an object 253
minimum reserved memory 362
monitoring
    cluster 30
database connection 11
document cache 466
job 55
machine performance 353
MicroStrategy system 639
report cache 466
user connection 68
moving a configuration object 258

N

named user license
    managing 225

setting when to verify 223
verifying 222
Narrowcast Server
    and deleted user 67
    application design 418
    connecting to Intelligence Server 419
    information source 419
    integrating with MicroStrategy Web 815
    system resource use and 418
Narrowcast Server system check in Health Center 731
network
    bandwidth 350
    monitoring 351
    performance considerations 348
    recommended configuration 350
    network address of the temp client 56, 68
node defined on 422
notification for failed transmission 572

O

object browsing defined on 44
    job processing steps 43
object cache defined on 517
    deleting 520
    disabling 518
    limiting memory usage 519
    matching algorithm 518
    summary of settings 521
Object Manager 256
    Access Control List 273
    child dependency 267
    copying an object 253
    international settings 268
    parent dependency 264
    Project Merge versus 254
    resolving a conflict 269
saving a dependent object | 265
starting | 261
table migration | 274
timestamp | 267
OLAP Services and system performance | 398, 407
Oracle 8i | 13, 776
orphan session | defined on 665

P

page-by and system performance | 406
parent dependency | defined on 264
partial idle project mode | 35
partitioned fact table | 110
example | 218
passthrough execution | 93
example | 217
PDF, exporting as | 54
performance counters and diagnostics | 744
Performance Monitor | 353
permissions | defined on 69
assigning | 70
list of | 73
server object | 74
use and execute | 77
viewing | 66
personal page execution | defined on 418
personal report execution | defined on 418
personalized drill path | 78
physical disk utilization | 354
physical warehouse schema | defined on 5
plain text, exporting as | 53
preferences, Web | 804
primary server | 29
prioritizing database connection thread | 390
private bytes | defined on 356
privileges | defined on 79
assigning | 79
denying | 80
governing with | 377
granting | 80
inheriting | 82
merging in User Merge | 114
processing results | 394
processor affinity | 230
project | defined on 7
cache viewing | 467
change journaling | 30, 315
changing status of | 35
comparing | 304
comparing reports between | 613
copying | 241, 246
default options for MicroStrategy Web | 804
deleting change journal entries | 322
duplicating | 241, 246
enabling change journaling | 315
governing summary | 412
idling | 30, 36
loading | 30, 35
locking | 255
logging changes | 315
merging | 296, 297
MicroStrategy Web default options | 804
purging a cache | 30
renaming | 661
resuming | 30, 36
SSD information | 756, 758
state information in the SSD | 758
testing | 611
tracking | 307
unloading | 30, 35
unlocking | 255
viewing changes 318
Project Comparison Wizard 304
comparing objects between two projects 305
project comparison process 305
using 305
project documentation in Enterprise Manager 672
project failover support 448
Project Merge 296
command 300
conflict resolution 303
Object Manager versus 254
project merge process 297
resolving a conflict 303
scheduling with Windows AT command 302
using 299
Project Merge Wizard 298
conflict resolution 303
merging projects 299
resolving a conflict 303
XML 254
project mode 31
execution idle 33
full idle 34
loaded 32
partial idle 35
request idle 33
unloaded 32
warehouse execution idle 34
Project Monitor 29
accessing 29
scheduling maintenance 30
using 29
project schema
exporting 30
printing 30
project source defined on 7
change journal
disabling 316
enabling 316
purging 322
change journaling 316
project status 31
changing 35
Project view
accessing 29
using 29
Project view of System Maintenance Monitor 29
project-versus-project integrity test 613
prompt
Command Manager and 606
Integrity Manager and 621
resolving 42
system 107
system performance and 406
prompted report in Integrity Manager 621
Public user group 64
purging
change journal 322
change journal for a project source 322
History List messages 504
result caches 474
statistics 336
Q
Query Builder, deleting managed objects 309
R
real-time analysis and Enterprise Manager 705
report
comparing
  from two baselines 613
  with a baseline 613
  with another report 611
cost 393
Enterprise Manager 669
instance 40
prompted in Integrity Manager 621
scheduling 553
size 395
system performance and 404
testing execution of 613
report cache
  configuring settings 476
deleting 468, 474
disabling for
  a project 466
  a single report 487
enabling for a single report 487
expiring 475
file format 462
history 460
index file 462
invalidating 468, 472
loading from disk 468, 471
location 461
managing 470
matching 460
matching algorithm 463
matching-history 460
memory usage 483
monitoring 466
project settings 478
scheduling an update 471
settings 476
status 468
types 460, 469
unloading to disk 468, 471
updating on a schedule 553
viewing 467
XML 461
Report Cache Monitor 468
report comparison tool 611
report complexity and system resource use 405
report cost defined on 393
report instance defined on 40
request
  governing 379
  types 38
request idle project mode 33
resolving a prompt 42
response file
  configuring 57
  installing 56
result cache defined on 456
  purging 474
  sharing options in cluster 427
result row limit troubleshooting 780
results
  delivery 399
  processing 394
results summary of integrity tests 631
resuming a project 30, 36
S
Sarbanes-Oxley change journaling 315
Scan MD tool 776
schedule defined on 547
  creating 547
  event-based 548
SSD information 758
time-triggered 548
scheduled e-mail, enabling with Web 815
scheduling defined on 545
  administration tasks 560
Command Manager script 596
project maintenance in Project Monitor 30
Project Merge to run 302
report 553
using the command prompt 657
schema object defined on 5
copying 258
script 595
execution errors 606
syntax errors 605
timeout errors 607
Search Export feature 307
secure sockets layer. See SSL.
security
assigning privileges 79
authentication 117
controlling application functionality 69
cookies 810
data access control 88
encryption 812
file-level 813
firewall 806
MicroStrategy Mobile 805
MicroStrategy Web 805
permissions 69
privileges 79
security role 82
setting up 61
single sign-on 62
SSL (HTTPS) 809
security filter defined on 95
managing with Project Monitor 30
merging multiple 104
metric level 98
range attributes 98
User Merge Wizard 115
security role defined on 82
user group and 81
User Merge Wizard 115
security view defined on 110
purpose 110
send now, enabling with Web 815
server definition defined on 6
Intelligence Server startup 15
permissions 74
SSD information 756, 757
server object defined on 74
server state dump (SDD) defined on 755
server state dump (SSD) contents 755
server, purging a cache in 31
Service Manager 18
Service Principal Name (SPN)
application server 188
IIS 186
Intelligence Server 181
setting
job priority 392
security 61
sharing caches 427, 430
silent installation 57
Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO) 194
single memory allocation governing 361
single sign-on 62
single-project integrity test 613
SPN. See Service Principal Name.
SPNEGO 194
SQL, limiting report’s SQL size 385
SSD. See server state dump.
SSL defined on 809
certificate 133
encryption 809
LDAP 132
<table>
<thead>
<tr>
<th>Index</th>
<th>System Administration Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>table</td>
</tr>
<tr>
<td></td>
<td>migrating with Object Manager 274</td>
</tr>
<tr>
<td></td>
<td>partitioning 110</td>
</tr>
<tr>
<td></td>
<td>task list defined on 40</td>
</tr>
<tr>
<td></td>
<td>technical support xxxvii</td>
</tr>
<tr>
<td></td>
<td>temp client 56, 68</td>
</tr>
<tr>
<td></td>
<td>testing</td>
</tr>
<tr>
<td></td>
<td>document execution 613</td>
</tr>
<tr>
<td></td>
<td>project 611</td>
</tr>
<tr>
<td></td>
<td>report execution 613</td>
</tr>
<tr>
<td></td>
<td>timestamp 331, 661</td>
</tr>
<tr>
<td></td>
<td>migrated object 267</td>
</tr>
<tr>
<td></td>
<td>time-triggered schedule 548</td>
</tr>
<tr>
<td></td>
<td>transmitter 566</td>
</tr>
<tr>
<td></td>
<td>best practices 572</td>
</tr>
<tr>
<td></td>
<td>deleting 576</td>
</tr>
<tr>
<td></td>
<td>email 574</td>
</tr>
<tr>
<td></td>
<td>failure notification 572</td>
</tr>
<tr>
<td></td>
<td>file 575</td>
</tr>
<tr>
<td></td>
<td>print 575</td>
</tr>
<tr>
<td></td>
<td>recommended tasks 572</td>
</tr>
<tr>
<td></td>
<td>Transmitter Editor 573</td>
</tr>
<tr>
<td></td>
<td>types 573</td>
</tr>
<tr>
<td></td>
<td>Transmitter Editor 573</td>
</tr>
<tr>
<td></td>
<td>triggering an event</td>
</tr>
<tr>
<td></td>
<td>automatically 551</td>
</tr>
<tr>
<td></td>
<td>manually 552</td>
</tr>
<tr>
<td></td>
<td>troubleshooting 738</td>
</tr>
<tr>
<td></td>
<td>data loading 664</td>
</tr>
<tr>
<td></td>
<td>LDAP 168</td>
</tr>
<tr>
<td></td>
<td>report result rows 780</td>
</tr>
<tr>
<td></td>
<td>tuning 341, 342</td>
</tr>
<tr>
<td></td>
<td>runtime capacity variables 344</td>
</tr>
<tr>
<td></td>
<td>system design 343</td>
</tr>
<tr>
<td></td>
<td>troubleshooting in LDAP 769</td>
</tr>
<tr>
<td></td>
<td>using with Web 809</td>
</tr>
<tr>
<td></td>
<td>standard authentication 120</td>
</tr>
<tr>
<td></td>
<td>example 218</td>
</tr>
<tr>
<td></td>
<td>starting</td>
</tr>
<tr>
<td></td>
<td>Intelligence Server 15</td>
</tr>
<tr>
<td></td>
<td>Intelligence Server...</td>
</tr>
<tr>
<td></td>
<td>statistics</td>
</tr>
<tr>
<td></td>
<td>analyzing 669</td>
</tr>
<tr>
<td></td>
<td>creating statistics tables 331</td>
</tr>
<tr>
<td></td>
<td>options 324</td>
</tr>
<tr>
<td></td>
<td>purging 336</td>
</tr>
<tr>
<td></td>
<td>setting the project to log 332</td>
</tr>
<tr>
<td></td>
<td>timestamp 331</td>
</tr>
<tr>
<td></td>
<td>troubleshooting 790</td>
</tr>
<tr>
<td></td>
<td>using in Enterprise Manager 639</td>
</tr>
<tr>
<td></td>
<td>status</td>
</tr>
<tr>
<td></td>
<td>cache 468</td>
</tr>
<tr>
<td></td>
<td>project 31</td>
</tr>
<tr>
<td></td>
<td>stopping Intelligence Server 16</td>
</tr>
<tr>
<td></td>
<td>subscriptions statistics logging 793</td>
</tr>
<tr>
<td></td>
<td>support</td>
</tr>
<tr>
<td></td>
<td>international xxxvi</td>
</tr>
<tr>
<td></td>
<td>support. See technical support.</td>
</tr>
<tr>
<td></td>
<td>synchronizing schedules for LDAP 158</td>
</tr>
<tr>
<td></td>
<td>system activity summary 679</td>
</tr>
<tr>
<td></td>
<td>System Administration Monitor 28</td>
</tr>
<tr>
<td></td>
<td>system architecture 1</td>
</tr>
<tr>
<td></td>
<td>configuring 347</td>
</tr>
<tr>
<td></td>
<td>system check 718, 723</td>
</tr>
<tr>
<td></td>
<td>manual 718</td>
</tr>
<tr>
<td></td>
<td>scheduled 719</td>
</tr>
<tr>
<td></td>
<td>System Maintenance Monitor</td>
</tr>
<tr>
<td></td>
<td>Cluster view 30</td>
</tr>
<tr>
<td></td>
<td>Project Monitor 29</td>
</tr>
<tr>
<td></td>
<td>Project view 29</td>
</tr>
<tr>
<td></td>
<td>system prompt defined on 107</td>
</tr>
<tr>
<td></td>
<td>system resources governing 352</td>
</tr>
<tr>
<td></td>
<td>system usage trend, analyzing 669</td>
</tr>
<tr>
<td></td>
<td>troubleshooting in LDAP...</td>
</tr>
<tr>
<td></td>
<td>using with Web...</td>
</tr>
<tr>
<td></td>
<td>standard authentication...</td>
</tr>
<tr>
<td></td>
<td>example...</td>
</tr>
<tr>
<td></td>
<td>starting...</td>
</tr>
<tr>
<td></td>
<td>Intelligence Server...</td>
</tr>
<tr>
<td></td>
<td>Intelligence Server...</td>
</tr>
<tr>
<td></td>
<td>statistics...</td>
</tr>
<tr>
<td></td>
<td>analyzing...</td>
</tr>
<tr>
<td></td>
<td>creating statistics tables</td>
</tr>
<tr>
<td></td>
<td>options...</td>
</tr>
<tr>
<td></td>
<td>purging...</td>
</tr>
<tr>
<td></td>
<td>setting the project to log</td>
</tr>
<tr>
<td></td>
<td>timestamp...</td>
</tr>
<tr>
<td></td>
<td>troubleshooting...</td>
</tr>
<tr>
<td></td>
<td>using in Enterprise Manager</td>
</tr>
<tr>
<td></td>
<td>status...</td>
</tr>
<tr>
<td></td>
<td>cache...</td>
</tr>
<tr>
<td></td>
<td>project...</td>
</tr>
<tr>
<td></td>
<td>stopping Intelligence Server</td>
</tr>
<tr>
<td></td>
<td>subscriptions statistics logging</td>
</tr>
<tr>
<td></td>
<td>support...</td>
</tr>
<tr>
<td></td>
<td>international xxxvi...</td>
</tr>
<tr>
<td></td>
<td>support. See technical support.</td>
</tr>
<tr>
<td></td>
<td>synchronizing schedules for LDAP</td>
</tr>
<tr>
<td></td>
<td>system activity summary...</td>
</tr>
<tr>
<td></td>
<td>System Administration Monitor</td>
</tr>
<tr>
<td></td>
<td>system architecture...</td>
</tr>
<tr>
<td></td>
<td>configuring...</td>
</tr>
<tr>
<td></td>
<td>system check...</td>
</tr>
<tr>
<td></td>
<td>manual...</td>
</tr>
<tr>
<td></td>
<td>scheduled...</td>
</tr>
<tr>
<td></td>
<td>System Maintenance Monitor</td>
</tr>
<tr>
<td></td>
<td>Cluster view...</td>
</tr>
<tr>
<td></td>
<td>Project Monitor...</td>
</tr>
<tr>
<td></td>
<td>Project view...</td>
</tr>
<tr>
<td></td>
<td>system prompt...</td>
</tr>
<tr>
<td></td>
<td>system resources governing</td>
</tr>
</tbody>
</table>

© 2011 MicroStrategy, Inc.
system requirements 343

U
UNIX, install LDAP SDK 131
unknown exception 755
unloaded project mode 32
unloading
  project 30, 35
  report cache to disk 468, 471
unlocking a project 255
update package defined on 277
updating a license 229
upgrading
  Enterprise Manager 659
  VLDB properties 13
usage trend analysis 669
use permission 77
user defined on 62
  anonymous/guest for LDAP 153
  creating 67
  deleting 67
  disconnecting 68
importing 67
importing an LDAP user 141
LDAP search filter 137
linking LDAP users 152
managing with Project Monitor 30
merging 113
profile 377
resources 374
  sharing an account 120
user address space defined on 356
User Connection Monitor 68
  viewing active user connections 68
user connection, viewing 68
user group defined on 63
  default 63
Guest 64
importing 67
LDAP 64
Public 64
  security role and 81
User Merge Wizard 113
ACL 114
  connection mapping 115
  group membership 114
  privileges 114
  process 114
  profile folder 114
  running 116
  security filter 115
  security role 115
user mode process dump 755
user preferences, Web 804
user session defined on 371
  governing 372
  information in the SSD 759
V
verifying a named user license 222
viewing
  change journal entries 318
  current database connection 11
  History List message details 504
  log file 752
  log file in Health Center 721
  results of an integrity test 631
virtual bytes defined on 357
virtual memory defined on 356
VLDB properties defined on 12
  governing summary 417
  upgrading 13
W

warehouse execution idle project mode 34
Windows AT command
  Command Manager 596
  Enterprise Manager data load 657
  Project Merge 302
Windows authentication
  example 217
  implementing 169
  troubleshooting 766
Windows Performance Monitor 353
  memory 357
Windows, linking login to LDAP user
  login 162
working set defined on 375
  governing 375

X

XML
  cache 461
  converting to HTML for Web 52
  delta and reports 375
  drill path
    governing 404
    personalized 78
  MCM and 361
  Project Merge Wizard 254
  results delivery and 399
  size 401
XML cache defined on 461
  Cache Monitor 467