1. Planning Your Installation ......................................................... 10
   Upgrade Considerations ....................................................... 11
   MicroStrategy Products and Components ............................. 11
   Installation Prerequisites ...................................................... 63
   Installation Considerations .................................................. 88
   Methods of Installation ........................................................ 105
   Licensing Information ......................................................... 110
   Installation and Configuration Checklists ............................ 110

2. Installing MicroStrategy on Windows ............................... 115
   Installation Procedure in Windows .................................... 117
   Configuring your MicroStrategy Installation ...................... 158

3. Installing MicroStrategy on Linux ................................. 160
   Installation Procedures on Linux ....................................... 161
   Configuring your MicroStrategy Installation ...................... 198
   Installing and Configuring MicroStrategy Library ............ 199
   MicroStrategy Library System Requirements ...................... 202
   Library Administration Control Panel ............................... 202
Installing MicroStrategy Library on Windows .................................. 208
Installing MicroStrategy Library on Linux .................................. 222
MicroStrategy Library Server Configuration Properties .............. 235
Configuring Collaboration Server ........................................... 240
How to Set Up Application Servers ......................................... 255
Linking MicroStrategy Library to MicroStrategy Web ................. 263
How to Enable Seamless Login Between Web, Library, and Workstation ................................................................. 264
Ensuring Images in MicroStrategy Web Work With Content in the Library ........................................................................ 267
MicroStrategy Library Web Internationalization ......................... 269
Load Balancer and Clustering Considerations for Library & Collaboration Server ......................................................... 272
Cluster the MicroStrategy Collaboration Server ......................... 279
Command Line Tools to Create Security Artifacts ..................... 284
Troubleshooting Your MicroStrategy Library Installation .......... 285
Certificate Store Integration with MicroStrategy Library .......... 302

4. Installing and Configuring MicroStrategy Identity ................. 310

Overview of Install ...................................................................... 311
MicroStrategy Identity Pre-Installation Instructions ................. 312
MicroStrategy Identity Installation Instructions ....................... 323
MicroStrategy Identity Post-Installation Instructions .............. 323
Troubleshooting Information .................................................... 333

5. Activating Your Installation .................................................... 340

Request an Activation Code ..................................................... 341
Activate Your Installation ......................................................... 344
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring your MicroStrategy Installation</td>
<td>345</td>
</tr>
<tr>
<td>Server Activation FAQ</td>
<td>346</td>
</tr>
<tr>
<td>6. Configuring and Connecting Intelligence Server</td>
<td>355</td>
</tr>
<tr>
<td>Communicating with Databases</td>
<td>356</td>
</tr>
<tr>
<td>Initial MicroStrategy configuration</td>
<td>370</td>
</tr>
<tr>
<td>Connecting to a Data Warehouse and Other Repositories</td>
<td>429</td>
</tr>
<tr>
<td>Creating a Project</td>
<td>449</td>
</tr>
<tr>
<td>Configuring your MicroStrategy Installation</td>
<td>450</td>
</tr>
<tr>
<td>Data Import with Proxy</td>
<td>450</td>
</tr>
<tr>
<td>7. Deploying MicroStrategy Web and Mobile Server</td>
<td>457</td>
</tr>
<tr>
<td>Deploying with IIS (Windows)</td>
<td>458</td>
</tr>
<tr>
<td>General Steps to Deploy MicroStrategy JSP Applications</td>
<td>462</td>
</tr>
<tr>
<td>Deploying with WebLogic and Apache (Solaris)</td>
<td>468</td>
</tr>
<tr>
<td>Deploying with WebSphere and IBM HTTP Server (AIX)</td>
<td>487</td>
</tr>
<tr>
<td>Deploying with Oracle Glassfish Server (Solaris)</td>
<td>498</td>
</tr>
<tr>
<td>Deploying with Tomcat (Windows)</td>
<td>509</td>
</tr>
<tr>
<td>Deploying with Tomcat (Linux)</td>
<td>517</td>
</tr>
<tr>
<td>Deploying with SAP NetWeaver (Windows)</td>
<td>523</td>
</tr>
<tr>
<td>Deploying with Oracle 10g (Windows)</td>
<td>527</td>
</tr>
<tr>
<td>Deploying with JBoss (Windows)</td>
<td>531</td>
</tr>
<tr>
<td>Administering Your MicroStrategy Web Deployment</td>
<td>537</td>
</tr>
<tr>
<td>Using Absolute Paths to Share Configuration Files</td>
<td>539</td>
</tr>
<tr>
<td>Configuring Third-Party Data Sources for Importing Data</td>
<td>540</td>
</tr>
<tr>
<td>Configuring your MicroStrategy Installation</td>
<td>543</td>
</tr>
</tbody>
</table>
8. Setting Up Documents and HTML Documents ...............544
   Prerequisites ......................................................545
   Executing Documents and HTML Documents in Linux ....546
   Configuring your MicroStrategy Installation ...............554
9. Automated Installation on Windows .............................555
   Installation Log File .............................................556
   Methods of Installation .........................................558
   Configuring your MicroStrategy Installation ...............622
10. Automated Installation on Linux ..................................624
    Silent Installation ................................................625
    Configuring MicroStrategy in Command Line Mode ........679
    Configuring your MicroStrategy Installation ...............680
11. Deploying OEM Applications .....................................681
    Creating DSNs for OEM Environments .......................684
    Configuring a MicroStrategy Installation ...................685
    Designing a Project and Reporting Environment ..........686
    Customizing MicroStrategy Web ...............................686
    Deploying a MicroStrategy OEM Application ................687
    Tuning an OEM Deployment ....................................694
    Updating OEM Applications ...................................695
    Creating a DSN for a Data Source ............................699
    Testing ODBC connectivity ....................................700
    Configuring MicroStrategy with a response.ini file .........703
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Adding or Removing MicroStrategy Components</td>
<td>735</td>
</tr>
<tr>
<td></td>
<td>Adding or Removing MicroStrategy Components on Windows</td>
<td>736</td>
</tr>
<tr>
<td></td>
<td>Re-installing MicroStrategy components on Windows</td>
<td>737</td>
</tr>
<tr>
<td></td>
<td>Uninstalling MicroStrategy Components on Windows</td>
<td>740</td>
</tr>
<tr>
<td></td>
<td>Uninstalling MicroStrategy Components on Linux</td>
<td>742</td>
</tr>
<tr>
<td>14</td>
<td>Export Engine Configuration</td>
<td>745</td>
</tr>
<tr>
<td></td>
<td>Installation of Export Engine</td>
<td>746</td>
</tr>
<tr>
<td></td>
<td>Export Engine: Puppeteer Integration</td>
<td>748</td>
</tr>
<tr>
<td></td>
<td>Export Engine Configuration Properties</td>
<td>752</td>
</tr>
<tr>
<td></td>
<td>Enable Local Resource Loading for the Export Engine</td>
<td>757</td>
</tr>
<tr>
<td>A</td>
<td>Connecting to Databases and Data Sources</td>
<td>759</td>
</tr>
<tr>
<td></td>
<td>Creating DSNs for Specific Data Sources</td>
<td>759</td>
</tr>
<tr>
<td></td>
<td>Creating Database Connections in Web</td>
<td>799</td>
</tr>
<tr>
<td></td>
<td>Configuring ODBC Parameters with ODBC.ini</td>
<td>802</td>
</tr>
<tr>
<td></td>
<td>MicroStrategy Hadoop Gateway Simplified Installation</td>
<td>804</td>
</tr>
<tr>
<td></td>
<td>Introduction to the MicroStrategy Hadoop Gateway</td>
<td>804</td>
</tr>
<tr>
<td></td>
<td>Environment Considerations</td>
<td>806</td>
</tr>
<tr>
<td></td>
<td>Using the MicroStrategy Hadoop Gateway</td>
<td>813</td>
</tr>
<tr>
<td></td>
<td>Troubleshooting</td>
<td>822</td>
</tr>
<tr>
<td></td>
<td>Known Issues</td>
<td>831</td>
</tr>
<tr>
<td></td>
<td>Frequently Asked Questions</td>
<td>833</td>
</tr>
</tbody>
</table>
Installing and Configuring Topology ................................................. 839

Introduction to Topology .................................................................. 840
Installing Topology ........................................................................... 842
Deploying MicroStrategy Library through a WAR File .................. 846
Enabling Topology Communication Through a Firewall ........... 847
Removing a Service from the Topology Monitor ....................... 848
Restarting the Consul ...................................................................... 849
Specifying the Topology Communication Agent ....................... 850
Changing the Monitoring Agent to a Communication Agent .... 852
Starting and Stopping Services ....................................................... 854
Supported MicroStrategy Platform Services ............................. 858
Troubleshooting Topology .............................................................. 874
Configuring Environments with Dynamic or Multiple IP Addresses ......................................................... 884
Certified Operating Systems for Topology ................................. 888
Certified Web Containers for Topology ........................................ 888

Installing and Configuring the Identity Messaging Service .. 890

Identity Messaging Service Dependencies ................................. 890
Installing the Identity Messaging Service on Linux .................. 891
Configuring the Identity Messaging Service ............................... 893
How to Monitor the Identity Messaging Service Deployment .. 897
Troubleshooting the Identity Messaging Service Installation .. 898
Identity Server Configuration to Work With Identity Messaging Service ................................................................. 898

19. MicroStrategy Identity Administration .................................. 901

Managing MicroStrategy Identity Administrators ...................... 902
Managing the Signing Certificate Authority ................................... 903
How to Install the Kerberos Authentication Service .......... 910
B.Troubleshooting ................................................................. 914
Reviewing General Installation Errors ................................. 914
Graph and Document Support of Non-Western European Fonts .......................................................... 915
Server Port Number Errors .................................................. 915
DSN Connection Errors ....................................................... 916
Metadata and Other Repository Creation Errors ............... 917
Permission Errors ............................................................... 918
PLANNING YOUR INSTALLATION
MicroStrategy business intelligence tools help organizations to monitor, report, and analyze all of their enterprise data. MicroStrategy helps you make decisions based upon the data within your organization's enterprise data warehouses and other business data sources.

An overview of the different MicroStrategy components and products is provided so that you can decide what you need to install. This includes details on supported functionality and describes important installation prerequisites that should be considered before you start installing MicroStrategy products.

The MicroStrategy products that you can install depend on your MicroStrategy license. Contact your MicroStrategy account executive with MicroStrategy licensing questions.

You can begin determining your installation and configuration plan by reviewing the following topics:

**Upgrade Considerations**

If you want to upgrade an earlier version of MicroStrategy products, see the Upgrade Guide before upgrading existing metadata.

**MicroStrategy Products and Components**

MicroStrategy has a range of products and components that you can install on different operating systems. Depending on the type of setup that you have, you can install various combinations of MicroStrategy components. The components described in this section offer a complete set of tools for creating, deploying, supporting, and maintaining your business intelligence applications.
MicroStrategy components and their subcomponents are described in relation to how the components are grouped together during the installation routine, as well as how they fit into MicroStrategy's product offerings.

**Memory Allocation for MicroStrategy Products and Services**

The tables below list the recommended available memory for MicroStrategy products and components to function properly as well as the at rest memory consumption of the related services.

### Intelligence Server

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Component</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 GB</td>
<td>Intelligence Server Module</td>
<td>MSTRSvr2_64 (MicroStrategy Intelligence Server)</td>
<td>50 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The non-sucking service manager (MicroStrategy Intelligence Server Log Consumer)</td>
<td>2 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Java ™ Platform SE binary</td>
<td>500 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The non-sucking service manager (MicroStrategy PDFExport Service)</td>
<td>2 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Java ™ Platform SE binary</td>
<td>525 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MJRefSvr_64</td>
<td>100 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MJHGoSMgr_64</td>
<td>1 GB</td>
</tr>
</tbody>
</table>
## Recommended Memory Components

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Component</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Universal Option</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Report Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLAP Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Distribution Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transaction Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multisource Option</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Clustering Option</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### MicroStrategy Web

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Component</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 GB</td>
<td>Web Server (ASP.NET)</td>
<td>IIS Worker Process</td>
<td>500 MB</td>
</tr>
<tr>
<td></td>
<td>Web Server (JSP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web Universal Analyst</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web Universal Professional</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web Universal Reporter</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Portlets</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GIS Connectors</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### MicroStrategy Office

⚠️ This information applies to MicroStrategy Office, the add-in for
Microsoft Office applications which is no longer actively developed.

It was substituted with a new add-in, MicroStrategy for Office, which supports Office 365 applications. The initial version does not yet have all the functionalities of the previous add-in.

For more information, see the MicroStrategy for Office page in the 2019 Update 1 Readme and the MicroStrategy for Office Online Help.

### Recommended Memory

<table>
<thead>
<tr>
<th>Component</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office Client</td>
<td>moipkg (Office Package Wizard)</td>
<td>10 MB</td>
</tr>
<tr>
<td></td>
<td>moicnfg (Office Configuration)</td>
<td>30 MB</td>
</tr>
<tr>
<td>Web Services for Office (ASP.NET)</td>
<td>IIS Worker Process</td>
<td>100 MB</td>
</tr>
<tr>
<td>Web Services for Office (JSP)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### MicroStrategy Mobile

<table>
<thead>
<tr>
<th>Component</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Server(ASP.NET)</td>
<td>IIS Worker Process</td>
<td>500 MB</td>
</tr>
<tr>
<td>Mobile Server(JSP)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### MicroStrategy Developer

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Component</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 GB</td>
<td>MSTRDesk</td>
<td></td>
<td>40 MB</td>
</tr>
<tr>
<td></td>
<td>MicroStrategy.XEG.WPFApp (Xquery Editor and Generator)</td>
<td></td>
<td>50 MB</td>
</tr>
<tr>
<td>Analyst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architect</td>
<td>MSTRDesk (MicroStrategy Architect - &lt;Project Name&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architect Function Plugin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server Administrator</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### MicroStrategy Messaging Services

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GB</td>
<td>Java™ Platform SE Binary (Apache Zookeeper)</td>
<td>75 MB</td>
</tr>
<tr>
<td></td>
<td>Java™ Platform SE Binary</td>
<td>100 MB</td>
</tr>
<tr>
<td></td>
<td>Java™ Platform SE Binary (Apache Kafka)</td>
<td>75 MB</td>
</tr>
<tr>
<td></td>
<td>Java™ Platform SE Binary</td>
<td>500 MB</td>
</tr>
</tbody>
</table>
# MicroStrategy Object Manager

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GB</td>
<td>ObjectManager</td>
<td>25 MB</td>
</tr>
<tr>
<td></td>
<td>ProjectMergeUI</td>
<td>25 MB</td>
</tr>
<tr>
<td></td>
<td>MARTT2UI</td>
<td>20 MB</td>
</tr>
<tr>
<td></td>
<td>MergeUtility</td>
<td>20 MB</td>
</tr>
</tbody>
</table>

# Command Manager

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GB</td>
<td>CmdMgrW</td>
<td>65 MB</td>
</tr>
</tbody>
</table>

# Enterprise Manager

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GB</td>
<td>MJMulPrc_64</td>
<td>10 MB</td>
</tr>
<tr>
<td></td>
<td>MSTREMService, MJMulPrc_64 (EM Service Running)</td>
<td>50 MB</td>
</tr>
</tbody>
</table>
## Narrowcast Server

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Component</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 GB</td>
<td>Narrowcast Administrator</td>
<td>MSTRNCAD (MicroStrategy Narrowcast Administrator)</td>
<td>25 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monitor (Monitor)</td>
<td>1 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MicroStrategy Logging Client</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MicroStrategy System Monitor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delivery Engine</td>
<td>MicroStrategy Distribution Manager</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MicroStrategy Execution Engine</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MicroStrategy Logging Consumer</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MicroStrategy Logging Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MicroStrategy NC PDF Formatter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subscription Portal</td>
<td>MicroStrategy SMTP Service</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Portal Administrator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subscription Portal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tutorial Delivery Installation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tutorial Delivery Configuration</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Analytics Module

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Component</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 GB</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

## Other Components

<table>
<thead>
<tr>
<th>Recommended Memory</th>
<th>Component</th>
<th>Services</th>
<th>Memory at Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 GB</td>
<td>SequeLink ODBC Socket Server</td>
<td>DataDirect SequeLink Agent (SLAgent55)</td>
<td>2 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataDirect SequeLink Server for ODBC Socket (SLSocket55)</td>
<td>2 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataDirect SequeLink Service Starter</td>
<td>1 MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataDirect Technologies Home</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ipexen (DataDirect Product Registration)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Microsoft Management Console (sladmin60)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Windows Command Processor (SequeLink Management Command Line Tool)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDX Cube Provider</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MySQL</td>
<td>mysqlid</td>
<td>450 MB</td>
</tr>
<tr>
<td></td>
<td>Apache Tomcat</td>
<td>Commons Daemon Service</td>
<td>300 MB</td>
</tr>
<tr>
<td><strong>Recommended Memory</strong></td>
<td><strong>Component</strong></td>
<td><strong>Services</strong></td>
<td><strong>Memory at Rest</strong></td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>--------------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Runner (Apache Tomcat 8.0 Tomcat8)</td>
<td></td>
</tr>
</tbody>
</table>

### Mandatory tools installed with each MicroStrategy Product

<table>
<thead>
<tr>
<th><strong>Component</strong></th>
<th><strong>Services</strong></th>
<th><strong>Memory at Rest</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Wizard</td>
<td>macfgwizw (Configuration Wizard)</td>
<td>50 MB</td>
</tr>
<tr>
<td>Diagnostics and Performance Monitoring Tool</td>
<td>MADPCfg (MicroStrategy Connectivity Wizard)</td>
<td>50 MB</td>
</tr>
<tr>
<td>License Manager</td>
<td>MALicMgrW_64 (MicroStrategy License Manager)</td>
<td>75 MB</td>
</tr>
<tr>
<td>Connectivity Wizard</td>
<td>MAMDCW (MicroStrategy Connectivity Wizard)</td>
<td>75 MB</td>
</tr>
<tr>
<td>Project Source Manager (Everything except Web, Mobile, Office)</td>
<td>MACONMAN</td>
<td>15 MB</td>
</tr>
</tbody>
</table>

### Tools installed with some MicroStrategy Products

<table>
<thead>
<tr>
<th><strong>Component</strong></th>
<th><strong>Services</strong></th>
<th><strong>Memory at Rest</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Source Manager (Everything except Web, Mobile, Office)</td>
<td>MACONMAN</td>
<td>15 MB</td>
</tr>
<tr>
<td>Listener Service</td>
<td>MSTRLsn2_64 (MicroStrategy Listener)</td>
<td>10 MB</td>
</tr>
<tr>
<td>Component</td>
<td>Services</td>
<td>Memory at Rest</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>(Intelligence Server, Enterprise Manager)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Listener (Everything except Web, Mobile, Office)</td>
<td>TestListener (MicroStrategy Test Listener)</td>
<td>10 MB</td>
</tr>
<tr>
<td>DB Query Tool (Everything except Web, Mobile, Office)</td>
<td>MADBQueryTool (MicroStrategy DB Query Tool)</td>
<td>50 MB</td>
</tr>
<tr>
<td>Cube Advisor (Everything except Web, Mobile, Office)</td>
<td>MstrCubeAdvisor (MicroStrategy Cube Advisor)</td>
<td>15 MB</td>
</tr>
<tr>
<td>Service Manager (Intelligence Server, Enterprise Manager, Narrowcast)</td>
<td>MASvcMgr_64 (MicroStrategy Service Manager)</td>
<td>80 MB</td>
</tr>
</tbody>
</table>

**MicroStrategy Web**

MicroStrategy Web is used by most business user roles. It offers an intuitive user interface instantly accessible from all major web browsers with no installation required. Business consumers can use Web to consume and interact with published scorecards, dashboards and reports. Power users benefit from extensive capabilities to create, design and modify analytics to be used by the business user community. Analysts will enjoy the all-inclusive set of self-service data discovery capabilities to blend data, explore visually and share insights.

The Web product also provides a plug-in for the Microsoft Office productivity suite that allows any user to inject analytics into business documents created in PowerPoint, Excel or Word, enabling these documents to contain the most up-to-date business data.
MicroStrategy Web Components

MicroStrategy implements Web using the .NET and JAVA technologies. This allows MicroStrategy Web to be deployed on Windows and Linux environments. For information on how to deploy MicroStrategy Web (ASP.NET) and MicroStrategy Web (JSP), see Chapter 7, Deploying MicroStrategy Web and Mobile Server.

MicroStrategy Web provides users with a highly interactive environment and a low-maintenance interface for reporting and analysis. Using the MicroStrategy Web interface, users can access, analyze, and share corporate data through any Web browser on any operating system. MicroStrategy Web provides ad hoc querying, industry-leading analysis, quick deployment, and rapid customization, making it even easier for users to make informed business decisions.

For steps to use the MicroStrategy Web reporting environment, refer to the online help in the MicroStrategy Web interface.

For information about configuring and tuning MicroStrategy Web, refer to the System Administration Guide.

MicroStrategy Web Versions

MicroStrategy Web is available in the following versions:

- **Web Reporter**: Business users are able to view all types of reports and scorecards and also personalize reports, print, drill, sort, export, choose between grid or graph format, and schedule or immediately send reports via email or to a file server or a printer.

- **Web Analyst**: This version provides all the functionality of Web Reporter plus the ability to drill anywhere, edit totals, pivot reports, add or remove fields from reports, create derived metrics, and create reports or ad hoc queries.
Web Professional: This full-featured version provides all the functionality of Web Analyst plus the ability to design scorecards, dashboards, and operational reports in design mode or WYSIWYG view mode. Web Professional users have advanced formatting capabilities as well as the ability to perform calculations across multiple data sources.

MicroStrategy Portlets

Though different portal products typically require different integration approaches, you can integrate MicroStrategy content and functionality into your portal using one of the out-of-the-box MicroStrategy Portlets. Each out-of-the-box MicroStrategy Portlet provides a full complement of portlet features that are not found in any single portal server product, and combines the most useful features of the portlet mechanisms currently available. These portlets are designed to take advantage of the storage and repository mechanisms of its particular portal product, without requiring users to make any adjustments or changes when implementing portlets within a portal.

MicroStrategy Portlets can embed folders, reports, documents, user History Lists, and a search page into the portals through easy-to-configure screens. The portlets provide the full range of OLAP manipulations such as sort, pivot, add subtotals, export, and add new calculations, as well as design functionalities such as changing the report display between grids and graphs, and toggling thresholds.

In portal environments, users are commonly already logged in and authenticated with the portal. This authentication can also be used to provide access to MicroStrategy Web within the portal without having to re-enter their login information. This process is known as single sign-on. Out-of-the-box MicroStrategy Portlets automatically include support for single sign-on.
For steps to install and configure out-of-the-box MicroStrategy Portlets for several major enterprise portal servers, see the Enterprise Portal Integration Help. This resource can be accessed from the MicroStrategy Product Manuals page, as described in Accessing manuals and other documentation sources, page xxiii.

MicroStrategy GIS Connectors

MicroStrategy Geospatial Information System (GIS) Connectors let you integrate with ESRI to create sophisticated GIS applications. GIS lets business users visualize data in forms such as maps, globes, reports, and charts so that they can identify and analyze relationships, patterns, and trends in their data.

For information on how to install and configure the MicroStrategy GIS Connectors, see the GIS Integration Help. This resource can be accessed from the MicroStrategy Product Manuals page, as described in Accessing manuals and other documentation sources, page xxiii.

MicroStrategy Office

This information applies to MicroStrategy Office, the add-in for Microsoft Office applications which is no longer actively developed.

It was substituted with a new add-in, MicroStrategy for Office, which supports Office 365 applications. The initial version does not yet have all the functionalities of the previous add-in.

For more information, see the MicroStrategy for Office page in the 2019 Update 1 Readme and the MicroStrategy for Office Online Help.

MicroStrategy Office lets every Microsoft Office user run, edit, and format any MicroStrategy report directly from within Microsoft applications such as Excel, PowerPoint, and Word. MicroStrategy Office is designed using Microsoft .NET technology and accesses the

MicroStrategy Office gives business users open and straightforward access to the full functionality of the MicroStrategy platform from familiar Microsoft Office applications. MicroStrategy Office serves as a Microsoft add-in, with MicroStrategy functionality exposed as a single toolbar in Microsoft Office applications.

To learn how to use MicroStrategy Office, refer to the MicroStrategy for Office Online Help and MicroStrategy Office online help.

MicroStrategy Office requires that MicroStrategy Web Services is also installed. For information on Web Services, see MicroStrategy Web Services (ASP.NET) and Web Services (J2EE), page 25.

Allowing Users to Install MicroStrategy Office from a Network Location

You can allow users to install MicroStrategy Office from a network location, as described in the procedure below:

1. Insert the MicroStrategy installation disk into the disk drive and close the MicroStrategy Main Menu window that opens automatically.

2. Browse to the Installations folder on the MicroStrategy installation disk.

3. Copy the Office folder and paste it to a network location of your choice.

To ensure that ASP.NET Framework and Web Services Enhancements (WSE) Runtime are installed on users' machines when they install MicroStrategy Office, copy the Utilities folder to the network location so that it is on the same folder level.
as the Office folder. The WSE Runtime is installed only if it is not already installed on the user’s machine.

4. Share the network location with any users who need to install MicroStrategy Office.

5. Notify MicroStrategy Office users to run either
   MicroStrategyOffice.msi or MicroStrategyOffice64.msi from within the Office folder to install MicroStrategy Office. These .msi files are for installing MicroStrategy Office on 32-bit and 64-bit versions of Microsoft Office, respectively. These users will need Microsoft Windows Installer 4.5 on their machine to install MicroStrategy Office.

   In addition to allowing users to install MicroStrategy Office from a network location, you can also use the

MicroStrategy Web Services (ASP.NET) and Web Services (J2EE)

MicroStrategy Web Services (ASP.NET) and Web Services (J2EE) are two options to support the use of MicroStrategy Office.

- MicroStrategy Web Services (ASP.NET) is an easy-to-deploy service. You can deploy the ASP.NET version using Microsoft IIS on a Windows environment.

- MicroStrategy Web Services (J2EE) provides a servlet-based version of MicroStrategy Web Services that is compatible with a Linux or Windows environment.
To support alternative ways to access the MicroStrategy business intelligence platform using the latest web services technologies such as ASP.NET, JNI, Java and Web protocols, such as Apache Axis, refer to the *MicroStrategy SDK, page 40* and the accompanying MSDL.

For information on deploying MicroStrategy Web Services ASP.NET and J2EE versions, refer to the MicroStrategy for Office Online Help.

**MicroStrategy Mobile**

MicroStrategy Mobile is an interactive interface of the MicroStrategy BI platform that lets mobile business users harness the analytical power of MicroStrategy through the use of their mobile devices. It’s the easiest, fastest, and most affordable way to mobilize analytics, and information-rich apps to an increasingly mobile and 24 x 7 workforce.

MicroStrategy Mobile and the MicroStrategy Mobile Server provide MicroStrategy reporting and analysis capabilities on Apple iOS and Android devices. MicroStrategy uses the intuitive interfaces of these mobile devices to let users explore information using touch and smart gestures. MicroStrategy Mobile Business Intelligence applications can support workflows that lead users through data to decisions.

MicroStrategy Mobile also provides application developers a new way to develop and deploy Mobile applications that is faster, easier, and more maintainable than using traditional Integrated Development Environments. MicroStrategy Mobile offers the following benefits:

- **Reduces the time to develop new Mobile applications:**
  MicroStrategy's Mobile application platform includes the infrastructure needed to support each new Mobile application, so that application developers only need to focus on creating the user experience and not on the back-end infrastructure.
• **Easy for non-developers to create professional Mobile applications**: MicroStrategy's Mobile applications do not require any coding. Using MicroStrategy's Mobile application platform, Mobile applications are assembled in a point-and-click fashion. Application designers can choose from an array of finely-designed displays and controls that are optimized for mobile devices.

• **Easy for companies to rapidly deploy Mobile application updates**: MicroStrategy's Mobile application platform uses an on-demand form of application deployment called "in-stream" deployment. As soon as new or updated applications are ready, they are instantly available to Mobile users directly from MicroStrategy's Mobile application platform.

• **One design for all devices**: MicroStrategy Mobile's ability to render the same application across different mobile device operating systems means less development time, less application management, and quicker support for a heterogeneous deployment of mobile devices.

To learn more about MicroStrategy Mobile, see the MicroStrategy Mobile Administration Guide and the MicroStrategy Mobile Design and Analysis Guide.

For information on how to deploy MicroStrategy Mobile Server (ASP.NET) and MicroStrategy Mobile Server (JSP), see Chapter 7, *Deploying MicroStrategy Web and Mobile Server*.

**MicroStrategy Server**

MicroStrategy Server benefits all user roles. The fully featured server infrastructure is the backbone of any MicroStrategy implementation and offers all the core platform services, which include:

• 64-bit server infrastructure to scale to big data volumes and a large number of users.

• Ability to connect to and join data from multiple data sources.
In-memory acceleration of analytical processing for instantaneous response.

Processing of all analytic styles from self-service data discovery to beautiful, immersive information apps to the industry's broadest spectrum of advanced analytics.

Proactive distribution of personalized reports and alerts.

Ability to embed actionable intelligence in analytical applications.

In addition to all the features above, the Server product includes highly useful monitoring and automation tools for organizations to effectively and efficiently manage their deployments.

MicroStrategy Intelligence Server

MicroStrategy Intelligence Server delivers world-class monitoring, reporting, and analysis on a single integrated platform, offering next generation BI capabilities for the full range of BI applications.

MicroStrategy Intelligence Server is the architectural foundation of the MicroStrategy platform. It performs the following critical tasks for the MicroStrategy BI platform:

- Runs queries, performs calculations, and formats reports
- Significantly improves user-perceived query performance
- Efficiently manages thousands of end-user requests (jobs)
- Serves as a central point for the MicroStrategy metadata

Intelligence Server also provides a library of over 150 different sophisticated mathematical and statistical functions, which can be added to. See the Functions Reference for details about these functions.

All other products in the platform work in conjunction with Intelligence Server and benefit from its broad functionality.
The subcomponents of MicroStrategy Intelligence Server are as follows:

- *MicroStrategy OLAP Services, page 29*
- *MicroStrategy Report Services, page 30*
- *MicroStrategy Distribution Services, page 31*
- *MicroStrategy Transaction Services, page 32*
- *MultiSource Option, page 32*

- Clustering Option, which allows you to cluster a group of Intelligence Server machines (up to four Intelligence Server machines) to take advantage of the many benefits available in a clustered environment.

  For information on clustering Intelligence Servers, see the System Administration Guide.

**MicroStrategy OLAP Services**

MicroStrategy OLAP Services uses the concept of Intelligent Cube, an in-memory version of a report that can be manipulated by the Analytical Engine. MicroStrategy Developer, MicroStrategy Web, and MicroStrategy Office users can slice and dice data in reports within the Intelligent Cubes without having to re-execute SQL against the data warehouse.
For information on OLAP Services, see the In-memory Analytics Guide.

MicroStrategy Report Services

MicroStrategy Report Services is the enterprise reporting engine of the MicroStrategy business intelligence platform. A MicroStrategy Report Services document contains objects representing data coming from one or more reports, as well as positioning and formatting information. It is used to format data from multiple reports in a single display of presentation quality.
For information on Report Services, see the Document Creation Guide and the Dashboards and Widgets Creation Guide.

MicroStrategy Distribution Services

MicroStrategy Distribution Services provides high-volume, automated distribution of reports, documents, dossiers, and business performance alerts via email, file servers, FTP servers, and networked printers.
MicroStrategy Transaction Services

MicroStrategy Transaction Services lets you embed write-back functionality into reports and dashboards for the purposes of decision-making or initiating a transaction. These transactions can include one-click approvals and denials, notes for tracking and directing business activity, and write-back to data sources in real time.

Users of MicroStrategy Web, MicroStrategy Mobile for iPhone, and MicroStrategy Mobile for iPad can employ these transaction capabilities from reports, dashboards, and MicroStrategy Mobile applications.

MultiSource Option

With MultiSource Option, you can connect a project to multiple relational data sources. This allows you to integrate all your information from various databases and other relational data sources into a single MicroStrategy project for reporting and analysis purpose. All data sources included using the MultiSource Option are integrated as part of the same relational schema for a project.
For information on using MultiSource Option, see the Project Design Guide.

**MicroStrategy Command Manager**

MicroStrategy Command Manager is an application designed to simplify and automate administration tasks, such as add, delete, or update enterprise-level data associated with large numbers of users and user groups. Additionally, Command Manager allows you to manage various configuration settings within the MicroStrategy platform.
For information on using Command Manager, see the System Administration Guide.

MicroStrategy Enterprise Manager

MicroStrategy Enterprise Manager provides insights about governing and tuning all areas of your MicroStrategy environment. With Enterprise Manager, you can see a variety of Intelligence Server usage statistics. The statistics shown in predefined reports displayed by Enterprise Manager can help you make scheduling decisions, analyze bottlenecks, and tune performance.
For information on using Enterprise Manager, see the Enterprise Manager Guide.

**MicroStrategy System Manager**

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

System Manager lets you create a workflow visually, allowing you to see the step-by-step process that leads the workflow from one configuration to the next. This visual approach to creating a workflow can help you to notice opportunities to troubleshoot and error check configurations as part of a workflow.

For information on using MicroStrategy System Manager to configure and deploy your MicroStrategy environments, see the System Administration Guide.

**MicroStrategy Narrowcast Server**

MicroStrategy Narrowcast Server proactively distributes personalized information to employees, business partners, and customers through a variety of devices, including mobile phones, email, and Web pages. The distribution of personalized messages and targeted offers is triggered according to predefined schedules and exception criteria, delivering information in a timely and convenient manner. Narrowcast Server also provides a self-subscription portal, easing administrative responsibilities and empowering information consumers to choose the information they receive. Narrowcast Server can draw information from relational or non-relational sources.
Subscriptions can also be supported through Intelligence Server with the introduction of Distribution Services. For information on Distribution Services, see *MicroStrategy Distribution Services, page 31*.

For information on Narrowcast Server subcomponents, see the *MicroStrategy Narrowcast Server Installation and Configuration Guide*.

**SequeLink ODBC Socket Server**

SequeLink is a complete, end-to-end solution for configuring and managing data access across virtually any number of data stores, operating systems, and deployment options. SequeLink ODBC Socket Server is required to support MicroStrategy Narrowcast Server. It can also be used to access Microsoft Access databases and Microsoft Excel files stored on a Windows machine from an Intelligence Server hosted on a Linux machine (see *MicroStrategy ODBC Driver for SequeLink, page 771*).

The SequeLink ODBC Socket Server that is provided with a MicroStrategy installation is for exclusive use with the MicroStrategy Product Suite. You are not licensed to use this product with any application other than MicroStrategy products. You can contact Progress® DataDirect® to purchase the SequeLink ODBC Socket Server for use with non-MicroStrategy products.

**MicroStrategy Architect**

MicroStrategy Architect is designed to meet the needs of application architects and developers. It includes all the schema development, change management, and modeling tools that enable architects to manage the full development life cycle of MicroStrategy applications. The Architect product allows IT organizations to flexibly share and distribute roles and responsibilities for development, testing,
promotions, and migrations during the application lifestyle, leading to vast improvements in organizational efficiency.

**MicroStrategy Developer**

MicroStrategy Developer provides analytical features designed to facilitate and perform the deployment of reports. It governs application objects such as reports, filters, and metrics.

Developer also enables you to create application objects. The application objects are built on the schema objects that are created in MicroStrategy Architect. These application objects are used to analyze and provide insight into relevant data. The following sections provide a brief description of the subcomponents for these products.

The subcomponents of MicroStrategy Developer include:

- MicroStrategy Analyst is a simplified version of MicroStrategy Developer, providing the basic interactive reporting functionality required by managers.

- MicroStrategy Developer is a full-featured version for power analysts and application developers. With a full range of analytical functionality, a rich library of functions, and intelligent workflow, Developer is well suited for power users.

- MicroStrategy Architect provides project designer functionality such as attribute, fact, hierarchy, and project creation and modification. Architect contains the following subcomponents:

  - MicroStrategy Function Plug-in Wizard is an add-in to the Microsoft Visual C++ compiler, which comes with a standard MicroStrategy installation. It allows you to create a C++ project, with which you can implement your own custom MicroStrategy function plug-in. The option to install this component is enabled only if Microsoft Visual C++ version 2005 (8.0) or version 2010
(10.0) is present on the system where the installation is being performed.


- MicroStrategy Server Administrator is a MicroStrategy Intelligence Server administrative console that provides functionality such as system monitoring, cache management, and user and group management.

When installing MicroStrategy Developer, your license key must be licensed for MicroStrategy Intelligence Server to install and access MicroStrategy Server Administrator.

For information on various options present in Developer to create and run reports, see the Basic Reporting Guide. After you are familiar with basic Developer concepts, see the Advanced Reporting Guide for information on advanced Developer functionality.

**MicroStrategy Architect**

MicroStrategy includes a project design tool known as Architect. Architect allows you to define all the required components of your project from a centralized interface. Architect also provides a visual representation of your project as you create it, which helps to provide an intuitive workflow.

For information on using Architect to design a project in MicroStrategy, see the Project Design Guide.

**MicroStrategy Integrity Manager**

MicroStrategy Integrity Manager is an automated report comparison tool. Report SQL, report data, and graphs can be executed and
compared in Integrity Manager to help customers verify change success. In addition, the report comparison output can be analyzed at the report level in MicroStrategy Integrity Manager, and as HTML and XML summary files that are generated to provide easily distributed results to other users.

To learn more about MicroStrategy Integrity Manager, see the System Administration Guide.

MicroStrategy Object Manager

MicroStrategy Object Manager provides complete life cycle management capabilities for MicroStrategy environments. Using Object Manager, you can copy objects within a project or across related projects.
For information on using Object Manager, see the System Administration Guide.

MicroStrategy SDK

The MicroStrategy SDK is a collection of programming tools, utilities, documentation, and libraries of functions or classes designed to customize and extend MicroStrategy products for integration within other applications. The programming tools provided by the MicroStrategy SDK—including programming instructions, points of access, and guidelines for developers—allow programmers to enhance the operation of their software by customizing and embedding the MicroStrategy BI platform.

The MicroStrategy SDK and MicroStrategy Developer Library (MSDL) are not included in the MicroStrategy installation. You can download the MicroStrategy SDK and access the MicroStrategy Developer Library from the MicroStrategy support site.
The MicroStrategy SDK provides access to the entire MicroStrategy platform and includes everything required to build a feature-filled business intelligence-enabled application. The MicroStrategy SDK is made up of the following components:

- The MicroStrategy SDK includes the following individual SDKs, which are described in detail in the MicroStrategy Developer Library:
  - Web SDK
  - MicroStrategy REST API
  - Visualization SDK
  - MicroStrategyLibrary SDK
  - Embedding SDK
  - Mobile SDK
  - Narrowcast Server SDK
• Intelligence Server SDK
• MicroStrategy Office SDK

• Each of the individual SDKs listed above is made up of some of or all the following components:
• A comprehensive set of APIs that includes:
  • REST API services
  • COM-based client-server API
  • XML-based Web API with support for Java/COM
  • Narrowcast Server API

The set of MicroStrategy APIs provides support for a variety of development environments, including Java, C++, VB, XML, and standard Web and client-server technologies.

• A complete set of SDK documentation for all the MicroStrategy products that includes:
  • Reference guides such as Javadocs for the APIs.
  • The MicroStrategy Developer Library (MSDL), which provides all the information required to understand and use the MicroStrategy SDK.

• A variety of development tools that include:
  • Source code and sample application code for typical customization tasks
  • Development tools and production-ready utilities to reduce code creation/maintenance and help you build customized applications.
  • Specialized development tools, such as the Portal Integration Kit and the Web Services Development Kit
• Features for packaging your application, including embedded (silent) installation, project mover for project maintenance and upgrade, and schema services to upgrade the metadata.

MicroStrategy Sample Projects

MicroStrategy provides a set of packaged analytic components built using the MicroStrategy platform. These include the Human Resources Analysis Module and the MicroStrategy Tutorial.

Human Resources Analysis Module

The Human Resources Analysis Module contains sample dossiers and reports, as well as the reporting objects that can be used to create typical Human Resources reports. The Human Resources Analysis Module Reference is a guide that provides sample usages and descriptions for each of the module's dossiers and reports and the supporting objects that define them.

The Human Resources Analysis Module can be mapped to a different warehouse or used as a starter kit to develop custom applications. The module consists of a MicroStrategy project in a metadata, a reference guide, and a default data model.

MicroStrategy Tutorial Reporting

MicroStrategy Tutorial Reporting is a sample MicroStrategy project with a warehouse, and a set of demonstration dossiers, reports, and other objects, designed to illustrate the platform's rich functionality. The MicroStrategy Tutorial Reporting metadata is provided as part of the MicroStrategy Analytics Modules metadata.

The theme of the Tutorial project is a retail store that sells electronics, books, movies, and music. The key features include:
Five hierarchies: Customer, Geography, Products, Promotions, and Time. Each hierarchy can be viewed graphically through MicroStrategy Developer and MicroStrategy Web (through documents).

A large number of customers and items purchased.

Five reporting areas: Human Resources, Inventory, Financial, Product Sales, and Supplier.

Options to create reports from MicroStrategy Web or Developer focusing on a particular analysis area, such as Customer, Inventory, Time, Products, Category, Employee, or Call Center.

For more information on the Tutorial project, refer to the Project Design Guide.

**MicroStrategy Identity**

MicroStrategy Identity is a multi-faceted enterprise solution that offers the best combination of security and convenience for accessing digital and physical assets. The first component of MicroStrategy Identity is MicroStrategy Badge; this product creates a mobile identity on your smartphone that replaces access control tools like keycards, passwords, and security tokens with a unified digital credential. The second component is MicroStrategy Communicator, which provides identity discovery, location awareness, and two-way communication. These features can be leveraged to coordinate activities across a distributed group of users enabled with MicroStrategy Badge.

Depending on your license key, you can choose to install:

- **MicroStrategy Identity Server**: synchronizes identities with Enterprise Management (IDM) systems of record. The MicroStrategy Identity Server accesses ID repositories through gateways and presents the identity to associated devices and systems.
- **MicroStrategy Identity Manager**: allows administrators to manage an organization's network of users, configure access to MicroStrategy Identity-enabled systems and resources, and distribute digital badges and keys.

- **MicroStrategy Communicator**: provides identity discovery, location awareness, and two-way communication. These features can be leveraged to coordinate activities across a distributed group of users enabled with MicroStrategy Badge.

Once you have installed one or more of these products, each user in your system can download the MicroStrategy Badge app to their mobile device. With this app users can validate their identities or access Identity-enabled systems and resources.

For installation instructions, see *Installing and Configuring MicroStrategy Identity*.

**MicroStrategy Messaging Services**

**MicroStrategy Messaging Services on Windows**

Messaging Services is a component that is coupled with the Intelligence Server during installations and upgrades. Messaging Services is configured out-of-the-box and runs automatically after the installation is completed.

After installation, you can see the following services are automatically started:

- **Apache Kafka** *(C:\Program Files (x86)\MicroStrategy\Messaging Services\Kafka\kafka_2.11-0.10.1.0)*
• **Apache ZooKeeper** (C:\Program Files (x86)\MicroStrategy\Messaging Services\Kafka\kafka_2.11-0.10.1.0)

• **MicroStrategy Intelligence Server Log Consumer** (C:\Program Files (x86)\MicroStrategy\Intelligence Server\KafkaConsumer)

By default MicroStrategy will still send Intelligence Server diagnostic logs to local disk. Diagnostic logs will be sent to the Messaging Services Server after you perform the following:

• **Enable MicroStrategy Messaging Services**

• **Turn On the Sending Log to Messaging Services Feature**

Afterwards you will see Kafka log files created in the Kafka installation folder:

C:\Program Files (x86)\MicroStrategy\Messaging Services\tmp\kafka-logs

Different Kafka topics will be created to store data for different MicroStrategy components.

Configuring Messaging Services after upgrading

By default, MicroStrategy Messaging Services are installed along with the Intelligence server upgrade.

Once you have completed the upgrade process, you need to enable MicroStrategy Messaging Services. If not, the Intelligence Server continues to write to the original log.
Messaging Services Workflow for Intelligence Server

- Intelligence Server is the Kafka Producer and can be deployed a single node or cluster.
- Kafka Server can be deployed as a single node or cluster.
- Intelligence Server Log Consumer can run on any machine that can be connected to a Kafka Server.

Enable MicroStrategy Messaging Services

Messaging Services configuration is saved in the MicroStrategy Intelligence Server configuration. It can be enabled or disabled on the fly, without restarting your Intelligence Server.

Command Manager Scripts for Messaging Services

To check if Messaging Services is enabled, execute:

```
LIST ALL PROPERTIES FOR SERVER CONFIGURATION;
```

To enable Messaging Services through Command Manager, execute:

```
ALTER SERVER CONFIGURATION ENABLEMESSAGINGSERVICES TRUE
```
CONFIGUREMESSAGINGSERVICES
"bootstrap.servers:10.15.208.236:9092/batch.num.messages:5000/queue.buffering.max.ms:2000";

In the example above set:

- **bootstrap.servers**: to your Kafka Server IP address and port number.
- **batch.num.messages**: to the number of messages to send in one batch when using asynchronous mode.
- **queue.buffering.max.ms**: to the maximum time to buffer data when using asynchronous mode.

You can specify more Kafka Producer configuration settings in this command following the same format.

Turn On the Sending Log to Messaging Services Feature
You can turn on the Sending Log to Messaging Services feature using either MicroStrategy Web or Command Manager.

From MicroStrategy Web

1. Log in using an Administrator account.
2. Open User Preferences > Project Defaults.
3. Locate **Sending Log to Messaging Services** in the Features for Customer Feedback section.
4. Select **On** from the drop-down menu.
5. Click **Apply**.
From Command Manager

1. Connect to your project source.

2. Execute the following:

   ```
   ALTER FEATURE FLAG "SENDING LOG TO MESSAGING SERVICES" ON;
   ```

Modifying Messaging Services Configuration

Apache Kafka Server

The Kafka Server can be configured by modifying the `server.properties` file found in:

```
C:\Program Files (x86)\MicroStrategy\Messaging Services\Kafka\kafka_2.11-0.10.1.0\config
```

Both Apache Kafka Server and ZooKeeper should be restarted after modifying the above configuration file.

Intelligence Server Log Consumer

By default the Log Consumer is connecting to the Local Kafka Server.

There are two ways to modify the configuration of Log Consumer:

1. **Delete `LogConsumer.properties` and execute the following command and follow the steps in the command line:**

   ```
   C:\Program Files (x86)\MicroStrategy\Intelligence
   ```
Server\KafkaConsumer>java -jar KafkaConsumer.jar

2. **Modify file** C:\Program Files (x86)\MicroStrategy\Intelligence Server\KafkaConsumer\LogConsumer.properties directly.

The default values after installation are:

```
folder_path=C:\\Program Files (x86)\\Common Files\\MicroStrategy\\Log\\DSSErrors
is_silent_mode=true  # indicate run consumer in silent mode
broker_port=9092  # Kafka Server port number
broker_hostname=127.0.0.1  # Kafka Server IP
poll_time_out=1000  # consumer connection time out
max_file_size_M=20  # max log file size in MB
max_num_bak=1  # number of backup files
```

**MicroStrategy Messaging Services Configuration for Clusters**

If you have clustered your Intelligence Servers and want to use a separate machine to run MicroStrategy Messaging Services after upgrading, complete the following steps for each node in the cluster.

- The minimum number of nodes for a cluster is **3**.

Each node must have the following installed:

- MicroStrategy Messaging Services
- Apache Kafka
Configure Zookeeper

1. **Browse to folder** C:\Program Files (x86)\MicroStrategy\Messaging Services\Kafka\kafka_2.11-0.10.1.0\config.

2. **Edit file** zookeeper.properties **by adding following lines:**

   ```
   clientPort=2181
   dataDir=C:\\Program Files (x86)\\MicroStrategy\\Messaging Services\tmp\zookeeper
   maxClientCnxns=0
   initLimit=5
   syncLimit=2
   server.1=10.27.20.16:2888:3888
   server.2=10.27.20.60:2888:3888
   server.3=10.15.208.236:2888:3888
   ```

   Each server parameter must contain a unique integer identifier as shown above. You attribute the server id to each machine by creating a text file named myid, one for each server, which resides in that server’s data directory, as specified by the configuration file parameter dataDir = C:\Program Files (x86)\MicroStrategy\Messaging Services\tmp\zookeeper

3. **Go to folder** C:\Program Files
4. Create a text file named `myid` containing the identifying value from the server parameter name in the `zookeeper.properties` file.

Configure Kafka

1. Browse to folder `C:\Program Files (x86)\MicroStrategy\MessagingServices\Kafka\kafka_2.11-0.10.1.0\config`.

2. Edit file `server.properties`, add a row
   ```
   zookeeper.connect=10.27.20.16:2181,10.27.20.60:2181,10.15.208.236:2181
   ```
   to the Zookeeper section.

3. Modify the `broker.id` value to a unique integer from other Kafka servers (the default value is 0), such as for node 10.27.20.60 we use number 2.
Start, Stop, Restart, and Check Status of Messaging Services

On Windows installations, open Task Manager > Services to start, stop, restart, and check the status of Messaging Services components.

MicroStrategy Messaging Services on Linux

Messaging Services is a component that is coupled with the Intelligence Server during installations and upgrades. Messaging Services is configured out-of-the-box and runs automatically after the installation is completed.

After installation, you can see the following services are automatically started:

- **Apache Kafka**
  /opt/mstr/MicroStrategy/install/MessagingServices/Kafka/kafka_2.11-0.10.1.0

- **Apache ZooKeeper**
  /opt/mstr/MicroStrategy/install/MessagingServices/Kafka/kafka_2.11-0.10.1.0

- **MicroStrategy Intelligence Server Log Consumer**
  /opt/mstr/MicroStrategy/install/IntelligenceServer/KafkaConsumer
By default MicroStrategy will still send Intelligence Server diagnostic logs to local disk. Diagnostic logs will be sent to the Messaging Services Server after you perform the following:

- **Enable MicroStrategy Messaging Services**
- **Turn On the Sending Log to Messaging Services Feature**

Afterwards you will see Kafka log files created in the Kafka installation folder:

```
/opt/mstr/MicroStrategy/install/MessagingServices/Kafka/tmp/kafka-logs
```

Different Kafka topics will be created to store data for different MicroStrategy components.

**Configuring Messaging Services after upgrading**

By default, MicroStrategy Messaging Services are installed along with the Intelligence server upgrade.

> Once you have completed the upgrade process, you need to enable MicroStrategy Messaging Services. If not, the Intelligence Server continues to write to the original log.

**Messaging Services Workflow for Intelligence Server**

- Intelligence Server is the Kafka Producer and can be deployed a single node or cluster.
- Kafka Server can be deployed as a single node or cluster.
- Intelligence Server log consumer can run on any machine that
can be connected to a Kafka Server.

Enable MicroStrategy Messaging Services

Messaging Services configuration is saved in the MicroStrategy Intelligence Server configuration. It can be enabled or disabled on the fly, without restarting your Intelligence Server.

Command Manager Scripts for Messaging Services

To check if Messaging Services is enabled, execute:

```
LIST ALL PROPERTIES FOR SERVER CONFIGURATION;
```

To enable Messaging Services through Command Manager, execute:

```
ALTER SERVER CONFIGURATION ENABLEMESSAGINGSERVICES TRUE CONFIGUREMESSAGINGSERVICES "bootstrap.servers:10.15.208.236:9092/batch.num.messages:5000/queue.buffering.max.ms:2000";
```

In the example above set:
• *bootstrap.servers*: to your Kafka Server IP address and port number.

• *batch.num.messages*: to the number of messages to send in one batch when using asynchronous mode.

• *queue.buffering.max.ms*: to the maximum time to buffer data when using asynchronous mode.

You can specify more Kafka Producer configuration settings in this command following the same format.

**Turn On the Sending Log to Messaging Services Feature**

You can turn on the Sending Log to Messaging Services feature using either MicroStrategy Web or Command Manager.

**From MicroStrategy Web**

1. Log in using and Administrator account.

2. Open User Preferences > Project Defaults.

3. Locate **Sending Log to Messaging Services** in the Features for Customer Feedback section.

4. Select **On** from the drop-down menu.

5. Click **Apply**.

**From Command Manager**

1. Connect to your project source.

2. Execute the following:

```
ALTER FEATURE FLAG "SENDING LOG TO MESSAGING SERVICES" ON;
```
Modifying Messaging Services Configuration

Apache Kafka Server

The Kafka Server can be configured by modifying the server.properties file found in:

/opt/mstr/MicroStrategy/install/MessagingServices/Kafka/kafka_2.11-0.10.1.0

Both Apache Kafka Server and ZooKeeper should be restarted after modifying the above configuration file.

Intelligence Server Log Consumer

By default the Log Consumer is connecting to the Local Kafka Server.

There are two ways to modify the configuration of Log Consumer:

1. **Delete the LogConsumer.properties file from**
   /opt/mstr/MicroStrategy/install/IntelligenceServer/KafkaConsumer, **execute the following command,** and **follow the steps in the terminal:**

   /opt/mstr/MicroStrategy/install/IntelligenceServer/KafkaConsumer java -jar KafkaConsumer.jar

2. **Modify file**

   /opt/mstr/MicroStrategy/install/IntelligenceServer/KafkaConsumer/LogConsumer.properties directly.

   The default values after installation are:
MicroStrategy Messaging Services Configuration for Clustered Environments

If you have clustered your Intelligence Servers and want to use a separate machine to run MicroStrategy Messaging Services after upgrading, complete the following steps for each node in the cluster.

The minimum number of nodes for a cluster is 3.

Each node must have the following installed:

- MicroStrategy Messaging Services
- Apache Kafka
- Apache Zookeeper

Configure Zookeeper

1. Browse to folder
   
   /opt/mstr/MicroStrategy/install/MicroStrategy/MessagingServices/Kafka/kafka_2.11-0.9.0.1/config.
2. Edit file `zookeeper.properties` by adding following lines:

```properties
maxClientCnxns=0
initLimit=5
syncLimit=2
server.1=10.27.20.16:2888:3888
server.2=10.27.20.60:2888:3888
server.3=10.15.208.236:2888:3888
```

Each server parameter must contain a unique integer identifier as shown above.

3. Go to folder

```
/opt/mstr/MicroStrategy/install/MicroStrategy
/MessagingServices/Kafka/kafka_2.11-0.9.0.1/tmp/zookeeper.
```

4. Create a file named `myid` containing the identifying value from the server parameter name in the `zookeeper.properties` file.

Configure Kafka

1. Browse to folder

```
/opt/mstr/MicroStrategy/install/MicroStrategy
/MessagingServices/Kafka/kafka_2.11-0.9.0.1/config.
```

2. Edit file `server.properties`, add a row
zookeeper.connect=10.27.20.16:2181,10.27.20.60:2181,10.15.208.236:2181 to the Zookeeper section.

```
#############################
Zookeeper
#############################
# Zookeeper connection string (see zookeeper docs for details).
# This is a comma separated host:port pairs, server. e.g. "127.0.0.1:3000,127.0.0.1:3001"
# You can also append an optional chroot string root directory for all kafka znodes.
# zookeeper.connect=localhost:2181
zookeeper.connect=10.27.20.16:2181,10.27.20.60:2181,10.15.208.236:2181
```

3. Modify the broker.id value to a unique integer from other Kafka servers (the default value is 0), such as for node 10.27.20.60 we use number 2.

```
#############################
Server Basics
#############################
# The id of the broker. This must be set to a
broker.id=2
```

Start, Stop, Restart, and Check Status of Messaging Services

Kafka Server and Zookeeper have been registered as service on Linux, so we can use service command to start, stop, and check status. The restart command is not supported.

To execute a service command for Kafka Server and Zookeeper, enter: /etc/init.d/kafka-zookeeper {start|stop|status}. 
To execute a service command for MicroStrategy Intelligence Server Log Consumer, enter: /etc/init.d/consumer-iserver {stop/start/status}.

MicroStrategy Platform Analytics

Platform Analytics is the next generation telemetry tool that leverages data from across the MicroStrategy platform including:

- Environment composition
- System usage
- Project usage
- User activity
- Content usage
- Cube usage
- Subscription usage
- Quality
- Licensing
- Identity

Data is streamed in real time through the MicroStrategy Messaging Services layer and stored in the Platform Analytics warehouse. Platform Analytics provides several ways to access, analyze, and act on this telemetry, including out of the box standard dossiers and native telemetry interfaces in MicroStrategy Workstation; empowering administrators to provide a better experience to MicroStrategy users.
Accessing Platform Analytics

You can access Platform Analytics data in three different ways depending on your needs:

- **By viewing the Platform Analytics data embedded in Workstation:** One of the exciting features of Platform Analytics is that it exposes some of the data that it captures directly in the user interface of Workstation. This allows users who would otherwise not know how to consume Platform Analytics data, to gain access to important MicroStrategy data. For more information, see How to View Dossier Usage in the Workstation Online Help.

- **By running the out-of-the-box Platform Analytics dossiers:** Platform Analytics ships with a MicroStrategy project that provides out-of-the-box dossiers designed to showcase some of the data that Platform Analytics captures for each of the different system areas. The dossiers included with Platform Analytics are:
  - **Compliance Telemetry:** Determine if a MicroStrategy implementation complies with the license entitlements.
  - **Cube and Cache Monitoring:** Ensure that cubes and caches are being fully leveraged to improve the performance of key analytics content.
  - **Error Analysis:** Detect errors and anomalies in the system and improve the experience of MicroStrategy users by fixing those issues.
  - **Object Telemetry:** Identify the most popular analytics content in the system and determine who is viewing it and how fast it runs.
  - **Project Overview:** Analyze the performance of the MicroStrategy projects and determine which users connect to them and which products they use to connect.
• **Subscription Analysis**: Determine which analytics content users subscribe to and how much load these subscriptions create on the system.

• **User Activity**: Monitor what users do in MicroStrategy and ensure that they have a positive experience free of performance issues and other errors.

• **By creating your own dossiers**: Platform Analytics also supports the creation of self-service content (dossiers, reports, and documents) which are based on the out-of-the-box schema and application objects included in the Platform Analytics project.

### Installation Prerequisites

Before you install MicroStrategy, you must have the following:

• MicroStrategy installation files.

• Before you begin upgrading any MicroStrategy systems, contact your MicroStrategy account executive to obtain a new license key for the version of software you are installing.

• License key from MicroStrategy.

• You can access the installation files by asking your system administrator to share the files on a network location.

• You can reduce the amount of data that has to be downloaded for an installation by excluding some of the installation files in the `Installations/DataFiles` folder. During installation, the MicroStrategy Installation Wizard then lists which of these files are required for your MicroStrategy installation. You can use this technique to provide only the files required to complete a MicroStrategy installation, which can then be used to reduce the amount of data packaged and downloaded for other MicroStrategy
installations in your organization. For steps to use this technique to create custom installation packages, see Creating Custom Installation Packages, page 106.

- Installation location for your MicroStrategy products

To install MicroStrategy, you must have the following permissions and privileges:

- **Windows**:
  - You must log on to your machine using a domain account with Windows administrative privileges for the domain or target machine.
  - The user installing MicroStrategy needs write permissions in the installation directory to complete the installation; otherwise the installation fails.

- **Linux**:
  - You need root access permissions for installation if you have purchased the CPU-based MicroStrategy license.
  - You need root access permissions to install MicroStrategy Identity components.

In addition to the information provided above, review the following sections before the installation:

- Recommended Installation Location and Example Deployments, page 65
- Hardware Requirements and Recommendations, page 68
- Software Requirements and Recommendations, page 78
Recommended Installation Location and Example Deployments

There are a countless number of possible arrangements for all the products available on the MicroStrategy platform, and what you decide to do depends largely on your installation environment and requirements. In general, though, the following recommendations are usually true for a typical business intelligence system:

- Intelligence Server should be installed on its own dedicated server machine.

- MicroStrategy Web should be installed on its own dedicated Web server machine.

- The rest of the products can be installed in varying combinations depending on who intends to use them and on what machines.

These are just suggestions to help you get started. Read the rest of this chapter for more detailed guidelines.

The following sections provide basic examples of differently sized production deployments with MicroStrategy products. The examples are generalized and do not include all of the MicroStrategy products. You can use these examples to help plan how to deploy MicroStrategy products.
Small production deployment

- 500 Web Users
- 50 MicroStrategy Office Users
- 100 E-mail, File/Print Server Users

MicroStrategy Web and Web Services

5 Users
MicroStrategy Developer

2 Users
MicroStrategy Architect

MicroStrategy Narrowcast Server

MicroStrategy Intelligence Server
MicroStrategy OLAP Services
MicroStrategy Report Services
MicroStrategy Distribution Services
Medium production deployment

Large production deployment
Hardware Requirements and Recommendations

MicroStrategy acknowledges that variables, such as CPU speed, CPU type, operating system version, service upgrades, file space, and physical and swap memory, are factors that play an important role in making your deployment of MicroStrategy a successful one.

Determining the necessary hardware requirements to support MicroStrategy is dependent on many factors including the complexity of your MicroStrategy environment, the deployment strategy of MicroStrategy features, user community requirements, expected peak usage requirements, and response time expectations. Factors such as these must be considered to determine the hardware requirements for your MicroStrategy production environment.

For details and exact information regarding supported and certified operating system versions for a MicroStrategy release, see the Readme on the MicroStrategy website.

For Linux systems, several system settings can affect the performance of MicroStrategy Intelligence Server. These settings do not need to be set before a MicroStrategy installation. For more information on these settings and their recommended values, see Recommended System Settings for Linux, page 103.

System hardware requirements and recommendations for Windows

The following table lists the recommended and minimum hardware requirements for MicroStrategy products. The information provided is intended to give you general guidance on hardware requirements to support the MicroStrategy product suite. Determining the necessary hardware requirements to support MicroStrategy is dependent on many factors including but not limited to the complexity of your MicroStrategy environment, the deployment strategy of MicroStrategy
features, user community requirements, expected peak usage requirements, and response time expectations. Factors such as these must be considered to determine the hardware requirements for your MicroStrategy production environment.

- To ensure the installation process is completed successfully, all MicroStrategy platform and Update installations require 15 GB of disk space for the installer itself. This is in addition to any component or common file storage requirements listed below.

- MicroStrategy installs a set of common files that are shared when installing multiple MicroStrategy products on the same machine. With a typical installation setup type, these files are installed on the C: drive.

In addition to the storage requirements listed for the products in the table below, you should estimate an additional 2 GB of storage space for the common files that are shared amongst all products. While this estimate is conservatively high, planning for this additional space helps to ensure a successful installation.

- The storage requirements listed in the table below for Intelligence Server and Narrowcast Server include additional space than is required for the initial installation. This additional space is to support the creation of the various files that these products require throughout their use in a MicroStrategy environment. Additional space may be required depending on the use of Intelligence Server and Narrowcast Server in your MicroStrategy environment.

- Intelligence Server is licensed based on CPU number and clock speed. Thus, Intelligence Server can only be installed on machines with a maximum clock speed that equals the licensed clock speed. If you try to install the product on a machine faster than what is licensed, installation fails. See the System Administration Guide for more information about licensing.
<table>
<thead>
<tr>
<th><strong>MicroStrategy Product</strong></th>
<th><strong>Processor</strong></th>
<th><strong>Memory</strong></th>
<th><strong>Storage</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroStrategy System Manager</td>
<td>x86 or x64 compatible</td>
<td>2 GB</td>
<td>0.5 GB</td>
</tr>
<tr>
<td>MicroStrategy Command Manager</td>
<td>x86 or x64 compatible</td>
<td>2 GB</td>
<td>0.25 GB</td>
</tr>
<tr>
<td>MicroStrategy Enterprise Manager</td>
<td>x64 compatible</td>
<td>1 GB</td>
<td>0.25 GB</td>
</tr>
<tr>
<td>MicroStrategy Object Manager</td>
<td>x86 or x64 compatible</td>
<td>1 GB</td>
<td>0.25 GB</td>
</tr>
<tr>
<td>MicroStrategy Developer products</td>
<td>x86 or x64 compatible</td>
<td>2 GB or higher</td>
<td>0.25 GB</td>
</tr>
<tr>
<td>MicroStrategy Intelligence Server</td>
<td>x64 compatible</td>
<td>4 GB or higher</td>
<td></td>
</tr>
</tbody>
</table>

Using 4 GB of RAM is a minimum level of support for the MicroStrategy Product Suite, which does not take into account the performance of a production system. Performance testing has Three times the amount of RAM available to Intelligence Server. For example, an Intelligence Server that is provided 4 GB of RAM requires 12 GB of hard drive space.
shown that 64 GB or more of RAM should be available to allow MicroStrategy Intelligence Server to fully support and take advantage of the complete feature set of the MicroStrategy Product Suite, while obtaining system-wide high performance. This level of system resources allows MicroStrategy Intelligence Server to fully use performance-improving technologies such as MicroStrategy OLAP.
<table>
<thead>
<tr>
<th>MicroStrategy Product</th>
<th>Processor</th>
<th>Memory</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroStrategy Integrity Manager</td>
<td>x64 compatible</td>
<td>2 GB or higher</td>
<td>0.25 GB</td>
</tr>
<tr>
<td>MicroStrategy Office</td>
<td>x86 or x64 compatible</td>
<td>2 GB</td>
<td>0.5 GB</td>
</tr>
<tr>
<td>MicroStrategy Mobile Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The MicroStrategy Mobile Server hardware requirements are the same as those for MicroStrategy Web Server hardware requirements.</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Narrowcast Server</td>
<td>x86 or x64 compatible</td>
<td>4 GB</td>
<td>4 GB</td>
</tr>
<tr>
<td>MicroStrategy SDK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The SDK is not included in the MicroStrategy installation and can instead be downloaded from the MicroStrategy support site.</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Analytics Modules</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>0.5 GB</td>
</tr>
<tr>
<td>MicroStrategy Web: Web Client</td>
<td>x86 or x64 compatible</td>
<td>2 GB or higher</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MicroStrategy Web can be accessed from a third-party web browser, which means there are no additional storage requirements.</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Web: Web Server</td>
<td>x64 compatible</td>
<td>4 GB or higher</td>
<td>0.5 GB</td>
</tr>
</tbody>
</table>
System hardware requirements and recommendations on Linux

The following information is intended to give you general guidance on hardware requirements to support the MicroStrategy product suite. Determining the necessary hardware requirements to support MicroStrategy is dependent on many factors including but not limited to the complexity of your MicroStrategy environment, the deployment strategy of MicroStrategy features, user community requirements, expected peak usage requirements, and response time expectations. Factors such as these must be considered to determine the hardware requirements for your MicroStrategy production environment.

- The storage recommendations listed in the table below provide an estimate for installing and supporting each MicroStrategy product on a separate machine. For information on the total size of a MicroStrategy installation when installing all MicroStrategy products on the same machine, see below.

- A successful configuration of Intelligence Server depends on a valid combination of an operating system and a CPU architecture. Valid operating system and CPU architecture combinations for Intelligence Server are listed in the table below.

- The storage requirements listed in the table below for Intelligence Server include additional space than is required for the initial installation. This additional space is to support the creation of the various files that these products require throughout their use in a MicroStrategy environment. Additional space may be required depending on the use of Intelligence Server in your MicroStrategy environment.
<table>
<thead>
<tr>
<th>MicroStrategy Product</th>
<th>Processor</th>
<th>Memory</th>
<th>Storage Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroStrategy System Manager</td>
<td>Linux: x86-64 compatible</td>
<td>2 GB or higher</td>
<td>3 GB on other Linux operating systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Command Manager</td>
<td>Linux: x86-64 compatible</td>
<td>2 GB or higher</td>
<td>3 GB on other Linux operating systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Intelligence Server</td>
<td>Linux: x86-64 compatible</td>
<td>16 GB or higher</td>
<td>Using 16 GB of RAM is a minimum level of support for the MicroStrategy Product Suite, which does not take into account the performance of a production system. Performance testing has shown that 64 GB or more of RAM should be available to allow MicroStrategy Intelligence Server to fully support and take advantage of the Three times the amount of RAM available to Intelligence Server.</td>
</tr>
<tr>
<td>MicroStrategy Product</td>
<td>Processor</td>
<td>Memory</td>
<td>Storage Recommendation</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>-------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>MicroStrategy Integrity Manager</td>
<td>Linux: x86-64 compatible</td>
<td>2 GB or higher</td>
<td>3 GB on other Linux operating</td>
</tr>
</tbody>
</table>
### MicroStrategy Product Specifications

<table>
<thead>
<tr>
<th>MicroStrategy Product</th>
<th>Processor</th>
<th>Memory</th>
<th>Storage Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroStrategy Web Services for Office</td>
<td>Linux: x86-64 compatible</td>
<td>2 GB or higher</td>
<td>3 GB on other Linux operating systems</td>
</tr>
<tr>
<td>MicroStrategy Mobile Server</td>
<td>The MicroStrategy Mobile Server hardware requirements are the same as those for MicroStrategy Web Server hardware requirements.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroStrategy SDK</td>
<td>The SDK is not included in the MicroStrategy installation and can instead be downloaded from the MicroStrategy support site.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Web: Web Server</td>
<td>Linux: x86-64 compatible</td>
<td>4 GB or higher</td>
<td>3 GB on other Linux operating systems</td>
</tr>
<tr>
<td>MicroStrategy Web: Web Client</td>
<td>Linux: x86-64 compatible</td>
<td>2 GB or higher</td>
<td>MicroStrategy Web can be accessed from a third-party web browser, which means there are no additional storage requirements.</td>
</tr>
</tbody>
</table>

### Storage Requirements for All MicroStrategy Products on Linux

The storage recommendations listed above provide the storage recommendations for the MicroStrategy products if they are installed individually on separate machines.

A conservative estimate of the total file size if you install all MicroStrategy products on the same machine, which can then share a set of common files, is 12 GB.
**Temporary directory requirements for installation**

In addition to the space requirements listed above, you also need free space in the temporary directory. When installing on Linux, the installer requires 100 MB of free space in the temporary directory. The default location of the temporary directory is `/tmp`.

If the space in the default temporary directory is inadequate, you can use the `tempdir` command line option to change the location of the temporary directory. This directory must already exist and it must be specified using its absolute path, for example:

```
./setup.sh -tempdir /home/user/tmp
```

If you change the location of the temporary directory, free space is still required in the default location of the temporary directory to launch the MicroStrategy installation routine.

**MicroStrategy Mobile hardware requirements for mobile devices**

The tables below list the MicroStrategy Mobile client application hardware requirements for various mobile devices. To verify updated requirement information, see the Readme.

**Flash memory**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Minimum</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash memory</td>
<td>32 MB</td>
<td>64 MB</td>
</tr>
</tbody>
</table>

**Android devices**

Devices with OS minimum 4.x and certain GPUs are certified. Refer to the third-party documentation for your Android device vendor to
determine the Graphics Processing Unit (GPU) for your device. To see the latest list of devices, see the current MicroStrategy Readme.

Software Requirements and Recommendations

See the MicroStrategy Readme for the specific software requirements and recommendations for MicroStrategy products on the Windows, UNIX, and Linux platforms.

Intelligence Server software requirements on Linux

For the exact information such as version numbers and space requirements, see the MicroStrategy Readme.

MicroStrategy Integrity Manager for Linux platforms has the same requirements as Intelligence Server. Therefore, you can use the information in this section for Intelligence Server and Integrity Manager requirements on Linux platforms.

The following MicroStrategy products require an X-windows-enabled environment on all Linux platforms:

- GUI-based MicroStrategy Installation Wizard
- Diagnostics and Performance Logging tool
- Service Manager

The following requirements also apply to all Linux platforms:

- A Web browser is required for viewing the MicroStrategy Readmes and online help.

- Windows Services for UNIX or Samba is required for HTML document support. Samba 3.0 is required for the support of HTML documents with alphanumeric names.

The requirements listed below describe general requirements as well as requirements specific to the UNIX and Linux platforms.
Be aware of the following before reviewing the sections listed above:

- The operating systems listed are deemed supported or certified to reflect the level of internal testing that each configuration was exposed to for the current release. MicroStrategy recommends using certified configurations over the supported configurations.

- MicroStrategy certifies and supports operating systems that are compatible with a set of CPU chipsets, referred to as CPU architectures, that are binary-compatible. MicroStrategy tests on at least one of the CPU chipsets within a set of binary-compatible CPU architectures for purposes of certifying and supporting operating systems with MicroStrategy products. A valid CPU architecture is provided in parentheses () to clarify the operating system software certified or supported for Intelligence Server.

- All Linux operating systems are 64-bit.

- For information on LDAP Servers certified and supported for LDAP authentication with various Intelligence Server machine environments, see the MicroStrategy Readme..

Configuring shared memory resources

To improve the performance, MicroStrategy Intelligence Server can be configured to use shared memory resources. To support this configuration, you must ensure that the Intelligence Server host machine uses values greater than or equal to the resource limits described below.

During installation you have the following options:

- **Exit the MicroStrategy setup wizard to do the required system changes (Recommended):** Select this option to cancel the installation and make the required system resource limit changes to support shared memory resources. This option is recommended for
production environments. Information on the recommended resource limits is below.

- **Allow the setup to reconfigure MicroStrategy to use Pipe as the Default IPC Mechanism**: Select this option to disable the use of shared memory resources for Intelligence Server, and instead use the pipe mechanism. Disabling the ability to use shared memory resources can decrease the performance of your MicroStrategy applications; therefore, this is not recommended for production environments.

- **Keep Shared Memory as the Default IPC Mechanism. (MicroStrategy may not work properly)**: Select this option to keep your system resource limits set at their current values to support shared memory resources. While this allows you to continue installation with the current system resource limits, Intelligence Server may not function properly after installation. If you plan to use shared memory resources for enhanced performance of your production environments, you should select the first option to exit the installation and make the required system changes.

The tables below provide recommended values for various system resource limits on Linux.

Modifying the system resource limits listed below can affect system-wide behavior and therefore, steps to modify these values are not given. You should refer to your Linux documentation and contact your system administrator to modify these settings.

Semaphores are used to synchronize shared memory communications. The names of the settings that control semaphores differ between operating systems as listed in the tables below:
<table>
<thead>
<tr>
<th>Setting Name on Linux</th>
<th>Description</th>
<th>Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>semmsl</td>
<td>Maximum number of semaphores in a semaphore set.</td>
<td>250</td>
</tr>
<tr>
<td>semmsns</td>
<td>Maximum number of semaphores in the system.</td>
<td>1024000</td>
</tr>
<tr>
<td>semopm</td>
<td>Maximum number of operations in a simple semaphore call.</td>
<td>250</td>
</tr>
<tr>
<td>semmni</td>
<td>Maximum number of semaphore sets.</td>
<td>4096</td>
</tr>
</tbody>
</table>

**Linux**

For Linux requirements, see the System Requirements in the *MicroStrategy Readme*.

**Supporting Intelligence Server memory allocation on Linux**

MicroStrategy recommends that the Linux kernel setting `vm.max_map_count` be defined as 5,242,880 bytes. This allows Intelligence Server to utilize system memory resources. If a lower value is used, Intelligence Server may not be able to use all available system resources. This can cause some Intelligence Server actions to fail due to lack of system resources, which could be completed if the additional system resources were made available by increasing the value for this kernel setting.

For information on this setting, including how to modify its value, refer to your third-party Linux operating system documentation.

**MicroStrategy Web JSP software requirements and recommendations**

To confirm the latest requirement information, see the *MicroStrategy Readme*. For specific patches, filesets, technology level, and other
requirements for UNIX and Linux operating systems, see *Intelligence Server software requirements on Linux, page 78.*

Web server software

For information on the exact version numbers, see the *MicroStrategy Readme.*

**MicroStrategy Web Services J2EE software requirements and recommendations**

To confirm the latest requirement information, see the *MicroStrategy Readme.*

**MicroStrategy SDK software requirements and recommendations for JSP environments**

The table below lists the JDK, JRE, and JVM requirements for the MicroStrategy SDK customizations for JSP environments. For complete MicroStrategy SDK software requirements, including .NET environment requirements, see the *MicroStrategy Readme.*

**MicroStrategy System Manager software requirements on UNIX/Linux**

For System Manager operating system requirements on Windows platforms, see the MicroStrategy *Readme.*

**MicroStrategy Command Manager software requirements on UNIX/Linux**

For Command Manager operating system requirements on Windows platforms, see the MicroStrategy *Readme.*
MicroStrategy Mobile software requirements for mobile devices

The tables below list the MicroStrategy Mobile client application software requirements for iPhone, iPod Touch, and iPad devices. To verify updated requirement information, see the MicroStrategy Readme.

MicroStrategy Mobile Server software requirements

The sections below list the MicroStrategy Mobile Server software requirements.

Mobile Server deployment requirements

- Mobile Server ASP.NET can be deployed using the same requirements listed for MicroStrategy Web (see the MicroStrategy Readme).

- Mobile Server JSP can be deployed using the same requirements listed for MicroStrategy Web (see MicroStrategy Web JSP software requirements and recommendations, page 81).

Web browsers for Mobile Server

For web browsers that are supported, refer to the MicroStrategy Readme.

Supporting IIS 7.0.x or IIS 7.5.x as a web server for MicroStrategy Web or Mobile Server

If you plan to use IIS 7.0.x or IIS 7.5.x as the web server for MicroStrategy Web or Mobile Server, you must ensure that some IIS options are enabled. The procedure below describes how to enable...
the options that are required to support IIS 7.0.x or IIS 7.5.x as a web server for MicroStrategy Web or Mobile Server.

To support IIS 7.0.x or 7.5.x as a web server for MicroStrategy Web or Mobile Server

The third-party products discussed below are manufactured by vendors independent of MicroStrategy, and the steps to configure these products is subject to change. Refer to the appropriate Microsoft documentation for steps to configure IIS 7.0.x or IIS 7.5.x.

1. On a Windows machine, open the Control Panel.
2. Double-click **Programs and Features**.
3. Click the **Turn Windows features on or off** task.
4. Expand **Internet Information Services**, and select the following options:
   a. Expand **Web Management Tools** and select:
      • **IIS Management Console**
      • **IIS Management Scripts and Tools**
      • **IIS Management Service**
   b. Expand **World Wide Web Services**, then expand **Application Development Features**, and select:
      • **.NET Extensibility**
      • **ASP.NET**
      • **ISAPI Extensions**
      • **ISAPI Filters**
c  Within World Wide Web Services, expand Common Http Features, and select:
   • Default Document
   • Static Content
d  Expand Security, and select:
   • Request Filtering
   • Windows Authentication

5  Click OK.

Platform Analytics Prerequisites

Before installing Platform Analytics on a Windows or Linux machine, ensure that all of the following prerequisites are met.

• The following ports must be open and available in the machines where you will install the Telemetry Server(s) and Platform Analytics:
  • 9092
  • 2181
  • 6379
  • 3306
  • 2888 and 3888 (only if you plan to cluster three or more Telemetry Servers)

• In Linux only:

  • The MySQL JDBC driver must be installed using rpm on the machine you will install Platform Analytics. Alternatively, you can copy the driver and rename the JAR file to mysql-connector-java.jar and place it under the /usr/share/java directory. The
MySQL JDBC driver must be version 8.0.12 or above.

- The MySQL ODBC driver must be installed on the Intelligence Server machines that will deploy the Platform Analytics project. In the creation dialog of the Connection Wizard, enter the full MySQL ODBC driver installation path in the driver field.

  For example, if the MySQL ODBC driver is installed on /usr/local/lib, enter the full /usr/local/lib/libmyodbc8w.so path in the driver field when creating the DSN. The MySQL ODBC driver must be version 5.6 or above.

- You must have a MySQL warehouse. Platform Analytics supports versions 5.6, 5.7, and 8.0.

  However, MySQL 8.0 is only certified to work with a database a user created with "Legacy Authentication Method". Platform Analytics does not support the option "Strong Password Encryption for Authentication" (SHA256 mode). SSL must be disabled in the MySQL warehouse. To disable SSL, run the following command to create a user:

  ```
  CREATE USER 'test'@'%' IDENTIFIED WITH mysql_native_password BY 'password' REQUIRE NONE;
  ```

  This MySQL warehouse can be installed in the machine where you will install Platform Analytics, or in any other machine provided that there is connectivity between the Platform Analytics machine and the MySQL machine through port 3306.

- Inside the MySQL warehouse you must create a database schema called platform_analytics_wh. Use the following syntax to create the Platform Analytics Repository:

  ```
  CREATE DATABASE `platform_analytics_wh` DEFAULT CHARACTER SET utf8
  ```
You must provide the credentials of a MySQL database user with the following permissions:

- Create
- Index
- Select
- Create/alter routine
- Insert
- Delete
- Create temporary tables
- Alter
- Update
- Create view
- Drop

For example, if the user is 'mstr' the command would be:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER,
CREATE TEMPORARY TABLES, CREATE VIEW, CREATE ROUTINE, ALTER ROUTINE
ON `platform_analytics_wh`.* TO 'mstr'@'%;
flush privileges;
```

If installing MySQL on another machine, the database user for the Platform Analytics Repository must be configured to allow remote access. Remote access can be enabled at the time of creating the DB User or after by following these steps:
1. Connect to the MySQL Server using any client (MySQL Workbench, DB Query Tool, etc.).

2. Run the command:

```
UPDATE mysql.user
SET Host='%' WHERE Host='localhost' AND User='<DB User Name>'; FLUSH PRIVILEGES;
```

Replace `<DB User Name>` with the user installing the Platform Analytics Repository.

The database user can now connect to this MySQL server instance from any remote machine.

- You must create a MicroStrategy user in the group **System Monitors** > **System Administrators** or have access to the default Administrator user.

- For an estimation of resource requirements for stable and performant operation of the Telemetry Store (previously called Platform Analytics Consumer) architecture under a consistent transactional load, see KB482872: Capacity Planning for Platform Analytics.

### Installation Considerations

The following section contains guidelines and considerations for installation.

### System Sizing Guidelines

The following topics describe sizing guidelines to consider when you initially set up MicroStrategy. You should periodically reevaluate the system and update it based on actual system performance and use.
This section describes only the most basic guidelines. For detailed information refer to the System Administration Guide.

Number of users

The number of users can be measured in the following ways:

- **Total users**: Users that are registered in the system. For example, if a corporate website is available to be viewed by 950 individuals, the site has 950 total users.

- **Active users**: Users that are logged into the system. If a site is available to be viewed by 950 total users and 30 of them are logged into the site, there are 30 active users.

- **Concurrent users**: Users that have jobs being processed by a server (MicroStrategy Web, Intelligence Server, and so on) at the same time. For example, a site is available to 950 total users, and 30 people are logged in. Of 30 active users, 10 have jobs being processed by the server simultaneously; hence there are 10 concurrent users.

Of these measures, the number of concurrent users is important to consider. Your system must support the maximum number of concurrent users you expect at any given time.

Report complexity

The more complex a report, the more Intelligence Server resources are required. In this context, a "complex" report is one that requires a lot of analytical processing. While reports with long, complicated SQL are certainly complex in nature, they do not necessarily require additional Intelligence Server resources to execute. It is the analytical processing in a report that creates additional stress on an Intelligence Server.
Since analytically complex reports create a heavier load on the Intelligence Server than simpler reports, you should have a general idea of what the average report complexity is for your system. Knowing this can help you decide on a caching strategy. For example, you may decide to pre-cache complex reports and determine the processing power your Intelligence Server needs.

The database server processes the SQL that Intelligence Server generates, so reports with extremely complex SQL can place additional stress on the database server. You should take this into account when sizing your database server machine.

**Ad hoc reports versus caches**

Report caches store the results of previously executed reports. If a client (MicroStrategy Web, Developer, and so on) requests a report that is cached, Intelligence Server returns the cached report results to the client. For any ad hoc reports that are not cached, Intelligence Server must go through the entire report execution cycle before it can return the results. For this reason, report caching allows better response time while minimizing the load on the Intelligence Server.

The benefits of caching are more apparent for complex reports than for simple reports. While caching a complex report may significantly improve execution time, a report cache for a simple report may not make much difference in this regard.

Therefore, the more complex the ad hoc reporting is in your system, the greater the overall load on the Intelligence Server. Be sure to take this into account when sizing your Intelligence Server machine.

The process for element browsing is similar to ad hoc reporting. Element browsing takes place when you navigate through hierarchies of attribute elements, for example, viewing the list of months in the year.
attribute. By default, caching is enabled for element browsing. In addition, you can limit the number of elements to be retrieved at a time.

Report Services document

Report Services documents utilize MicroStrategy objects to run complex and sophisticated reports. The datasets available to a document determine its content. Each dataset represents a report and its component objects, such as attributes, metrics, custom groups, and consolidations. When a dataset is available to a document, the entire report or any component object from that dataset can be included in the document output.

When creating a document, refer to the following guidelines to avoid an increase in the Intelligence Server execution time and the overall CPU usage:

- The datasets should be few in number, but large in size.
- The number of grids in the output document should be less in number. Consolidate the data to fit into fewer grids in the output document, where possible.
- Use of complex elements, such as consolidations, custom groups, and smart metrics can increase the Intelligence Server usage, especially if arithmetic operators are used in element definitions.
- Use Custom formatting only when required.

OLAP Services

OLAP Services store reports as Intelligent Cubes in the physical memory of the Intelligence Server. When these Intelligent Cubes are cached in memory, report manipulations, such as adding derived metrics and derived elements, formatting, and drilling within the Intelligent Cube, take considerably less time. This is the case because the new SQL is not run against the database.
OLAP Services provide enhanced report manipulation functionality at the cost of Intelligence Server resources, as the cubes are stored in the memory of the Intelligence Server. Consider the following factors to determine the size of the Intelligent Cubes:

- Intelligence Server resources
- Expected response time
- User concurrency

You must monitor Intelligence Server CPU utilization and memory usage closely as OLAP Services might have an impact on the performance of the platform, particularly the memory and report response time. For information on OLAP Services, see the In-memory Analytics Guide. Additional performance tuning best practices for OLAP Services are provided in the System Administration Guide.

Additional considerations

Numerous factors can affect system performance, most of them related to system specifics, which makes them difficult to predict. Listed below are items you should consider when determining the requirements for your system:

- Developer versus MicroStrategy Web usage—MicroStrategy products are designed with the assumption that the majority of users access the system through MicroStrategy Web while a smaller percentage use the Developer products.

- Statistics logging—Statistics logging is very useful for analyzing and further refining the system configuration based on actual usage. However, logging all statistics all the time can create a noticeable increase in system response time. For this reason, you might choose to log only a subset of the statistics generated or only log statistics periodically.
Backup frequency—Caches can be stored in memory and on disk. When you enable backup, you allow the Intelligence Server to write all cache files to disk. If the backup frequency is set to the default of zero, backup files are written to disk as soon as they are created. However, writing all cache files to disk all the time can cause a noticeable reduction in system performance.

Set the backup frequency to a value that minimizes disk writes and optimizes memory usage for your system.

Ratio of MicroStrategy Web servers to Intelligence Servers—In a typical system you should have a 1:1 ratio of Intelligence Servers to MicroStrategy Web servers. This ensures that resources on both sides are optimized. However, you might find it useful to add Intelligence Servers or MicroStrategy Web servers depending on your particular requirements.

Report Styles—MicroStrategy Web provides a set of different XSL report styles. These styles provide an easy way for you to customize how reports look. Due to the varying complexity of these styles, some might require more processing than others.

MicroStrategy Professional Services for high performance

MicroStrategy Professional Services has identified five primary levers customers can use to get dramatically faster performance:

- Employ in-memory Business Intelligence
- Design high performance dashboards
- Optimize query efficiency
- Implement effective caching strategies
- Configure MicroStrategy for high performance
In just one week, MicroStrategy Professional Services, will conduct a thorough examination of your Business Intelligence implementation, providing you with actionable recommendations on these five key areas to improve overall performance. The MicroStrategy Performance Analysis service delivers:

- **Performance optimization roadmap**: A customized report with prioritized recommendations to achieve performance goals.

- **System configuration**: Optimum configuration setting recommendations to achieve efficient use of resources across different MicroStrategy products.

- **Performance monitoring plan**: A set of performance related metrics to proactively monitor and identify performance opportunities.

To learn how MicroStrategy Professional Services can help you assess and prioritize your performance opportunities with a Performance Analysis, see [http://www.microstrategy.com/services-support/overview](http://www.microstrategy.com/services-support/overview).

**Common questions about sizing**

The sections below provide brief explanations to common sizing questions. For detailed information on tuning your MicroStrategy environment, see the *Tuning your System for Best Performance* chapter in the *System Administration Guide*. The sections below also provide other additional resources.

**Why should I increase the processor speed of Intelligence Server?**

Increasing the processor speed of Intelligence Server enhances performance and reduces execution time for all analytical tasks and for requests from the Extensible Markup Language (XML) and Component Object Model (COM) application programming interfaces
(APIs). If you see that the machine or machines are running consistently at a high capacity, for example, greater than 80%, it may be a sign that a faster processor would improve the system's capacity.

For more detailed information on tuning your processors for your MicroStrategy environment, see the section *Managing system resources* in the *System Administration Guide*.

**Why should I add more processors to Intelligence Server?**

Adding more processors to the Intelligence Server allows for a better load distribution among the processors. This provides an overall performance gain. If you notice that the processor is running consistently at a high capacity, for example, greater than 80%, consider increasing the number of processors.

For more detailed information on tuning your processors for your MicroStrategy environment, see the section *Managing system resources* in the *System Administration Guide*.

**Why should I increase memory on the machine that hosts Intelligence Server?**

If the physical disk is utilized too much on a machine hosting Intelligence Server, it can indicate that there is a bottleneck in the system's performance. To monitor this on a Windows machine, use the Windows Performance Monitor for the object *PhysicalDisk* and the counter `% Disk Time`. If you see that the counter is greater than 80% on average, it may indicate that there is not enough memory on the machine.

For more detailed information on tuning your machine’s memory for your MicroStrategy environment, see the section *Managing system resources* in the *System Administration Guide*. 
What would more network bandwidth do for me?

You can tell whether your network is negatively impacting your system's performance by monitoring how much of your network's capacity is being used. To monitor this on a Windows machine, 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.

For very large result sets, increasing network bandwidth reduces bottlenecks created by network congestion. The result is larger data flow and faster query response time.

For more detailed information on tuning your network for your MicroStrategy environment, see the section *How the network can affect performance* in the **System Administration Guide**.

How many CPUs can a user fully utilize?

One user can fully utilize up to one CPU, regardless of the number of CPUs available in the server. The load is split across multiple CPUs in multi-processor servers.

For more detailed information on how licensing can affect the utilization of CPUs, see the **System Administration Guide**.

What is the advantage of using hyper-threading for a dual processor?

The advantage of using hyper-threading with a dual processor is that it decreases the overall CPU usage. The use of hyper-threading is recommended if you have a large number of users.
What is the disadvantage of using hyper-threading for a dual processor?

The disadvantage of using hyper-threading is that it increases the Intelligence Server execution time slightly. Therefore, for faster processing, the use of hyper-threading is not recommended.

What is the largest Intelligent Cube size that I can store in an Intelligence Server?

Intelligent Cubes must be stored in Intelligence Server memory for reports to access their data. While this can improve performance of these reports, loading too much data onto Intelligence Server memory can have a negative impact on Intelligence Server’s ability to process jobs. For this reason, it is important to govern how much Intelligent Cube data can be stored on the Intelligence Server.

For information on governing Intelligent Cube memory usage, loading, and storage, see the System Administration Guide.

International Support

The following table lists the language selection possibilities for different installation cases:

<table>
<thead>
<tr>
<th>Installation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fresh installation on a system in which MicroStrategy application has never</td>
<td>The MicroStrategy Installation Wizard prompts you to select the language from the drop-down list.</td>
</tr>
<tr>
<td>been installed before</td>
<td>The user language in the product interface is the language that you select during installation.</td>
</tr>
<tr>
<td>Repair or maintenance installation on a system on which MicroStrategy</td>
<td>All subsequent executions of the installation routine are displayed in the language that you selected the first time you installed the product on the system.</td>
</tr>
<tr>
<td>application has been installed before</td>
<td>The user language in the product interface is also the language that you selected the first time you</td>
</tr>
</tbody>
</table>
Installation and Configuration Guide

<table>
<thead>
<tr>
<th>Installation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completely uninstalling all the MicroStrategy products and installing the same version or a newer version</td>
<td>installed the product on the system. If you uninstall all the products and install either the same version or a higher version again, the MicroStrategy Installation Wizard prompts you to select the language from the drop-down list. <strong>Note:</strong> Even if you select a language from the language prompt in the installation routine, it has no effect on the default language of the product interfaces.</td>
</tr>
</tbody>
</table>

During installation, the installation Online Help is displayed in English only.

**MicroStrategy Web and Intelligence Server Compatibility**

You must ensure the versions of MicroStrategy Web and Intelligence Server are compatible. For example, MicroStrategy Web 2019 can only connect to Intelligence Server 2019 or later. For a complete list of compatible MicroStrategy Web and Intelligence Server versions, refer to the *MicroStrategy Readme*.

Refer to the *MicroStrategy Readme* for the complete MicroStrategy platform compatibility and interoperability specification. In addition, you can contact MicroStrategy Technical Support for the latest information and updates.

**Certified ODBC Drivers for MicroStrategy Intelligence Server**

MicroStrategy certifies ODBC drivers for Windows and Linux for Intelligence Server and different DBMS types. MicroStrategy-branded ODBC drivers are installed with the MicroStrategy products.
For a complete list of certified and supported configurations with exact version numbers, refer to the certified and supported configurations listed in the Readme.

## Certificates Used During MicroStrategy Identity Installation and Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Example file path</th>
<th>Where certificates are used</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroStrategy Identity Server SSL/HTTPS Certificate</td>
<td>/certs/HTTPSServerCertificate.crt</td>
<td>(1) Within the MicroStrategy Installation Wizard, on the page for MicroStrategy Identity, the field labeled &quot;SSL Server Certificate&quot; or &quot;SSL Certificate file&quot;. &lt;br&gt; (2) Within the MicroStrategy Identity Configuration web interface, on the first page, the field labeled &quot;SAML Certificate&quot;.</td>
<td>(1) SSL certificate used by the Tomcat web services. &lt;br&gt; (2) SAML certificate provided by MicroStrategy Identity Manager.</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>/certs/HTTPSServerCertificate.ke</td>
<td>(1) Within the MicroStrategy Installation Wizard, on the page for MicroStrategy Identity, the field labeled &quot;SSL Server Certificate&quot; or &quot;SSL Certificate file&quot;.</td>
<td>(1) SSL</td>
</tr>
<tr>
<td>Name</td>
<td>Example file path</td>
<td>Where certificates are used</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Identity Server</td>
<td></td>
<td>MicroStrategy Installation Wizard, on the page for MicroStrategy Identity, the field labeled &quot;SSL Server Certificate Key&quot; or &quot;Private Key File&quot;. (2) Within the MicroStrategy Identity Configuration web interface, on the first page, the field labeled &quot;SAML Key&quot;.</td>
<td>private key for the certificate used by the Tomcat web services. (2) SSL private key for the SAML certificate provided by MicroStrategy Identity Manager.</td>
</tr>
<tr>
<td>SSL/HTTPS Certificate Private Key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Third-Party Certificate Authority Root Certificate</td>
<td>/certs/ca-root.crt</td>
<td>Prior to running the MicroStrategy Installation Wizard, include it in the MicroStrategy Identity Server Certificate Authority.</td>
<td>SSL certificate representing the Certificate Authority that signed the MicroStrategy Identity Server SSL/HTTPS Certificate. Each CA has 1</td>
</tr>
<tr>
<td>Name</td>
<td>Example file path</td>
<td>Where certificates are used</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MicroStrategy Identity Server Signing Certificate Authority Certificate</td>
<td>/certs/ushersigningca.crt</td>
<td>Prior to running the MicroStrategy Installation Wizard, include it in the MicroStrategy Identity Server Certificate Authority Chain (see below).</td>
<td>SSL certificate representing the Certificate Authority that signed the MicroStrategy Identity Server SSL/HTTPS Certificate. Each CA may have 1 or more intermediate certificates.</td>
</tr>
<tr>
<td>MicroStrategy Identity Server Signing Certificate Authority Certificate</td>
<td>/certs/ushersigningca.crt</td>
<td>Within the MicroStrategy Identity Configuration web interface, on the first page, the field labeled &quot;SSL Certificate Authority Certificate&quot;.</td>
<td>SSL certificate used by the MicroStrategy Identity Server to sign any Certificate Signing Request (MicroStrategy Badge, Directory Agent, Identity SDK-based apps,</td>
</tr>
<tr>
<td>Name</td>
<td>Example file path</td>
<td>Where certificates are used</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>-----------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>MicroStrategy Identity Server Signing Certificate Authority Private Key</td>
<td>/certs/ushersigningca.key</td>
<td>Within the MicroStrategy Identity Configuration web interface, on the first page, the field labeled &quot;SSL Certificate Authority Key&quot;.</td>
<td>SSL private key for the MicroStrategy Identity Server Signing CA.</td>
</tr>
<tr>
<td>MicroStrategy Identity Server Certificate Authority Chain</td>
<td>/certs/UsherCACchain.pem</td>
<td>Within the MicroStrategy Installation Wizard, on the page for MicroStrategy Identity, the field labeled &quot;CA Certificate Chain&quot; or &quot;SSL Certificate Chain file&quot;.</td>
<td>Certificate store that includes all of the Certificate Authority certificates that the Tomcat web services will trust. Includes the third-party CA and MicroStrategy Identity Server Signing CA.</td>
</tr>
</tbody>
</table>

This table does not include files used for MicroStrategy Identity components deployed outside of the MicroStrategy Identity Server, such as the Directory Agent, Physical Access Adapters, and third-party Web/Mobile applications implementing Identity APIs.
Recommended System Settings for Linux

Linux systems allow processes and applications to run in a virtual environment. This means that each process, depending on its owner and the settings for certain environment variables, are run using a distinct set of properties that affect how much memory the process can use, how many CPU seconds it can use, what thread model it can use, how many files it can open, and so on.

MicroStrategy Intelligence Server installs on Linux systems with the required environment variables set to ensure that the server's jobs are processed correctly. However, as mentioned above, some settings are related to the user who starts the process (also known as the owner of the process) and other settings can only be set by the system administrator. Some of these settings may also have limits enforced for reasons unrelated to supporting MicroStrategy.

The table below lists MicroStrategy's recommendations for system settings that can affect the behavior of Intelligence Server.

Modifying the system settings listed below can affect system-wide behavior and therefore, steps to modify these values are not given. You should refer to your Linux documentation and contact your system administrator to modify these settings.

<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Description</th>
<th>Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ulimit name</td>
<td>Maximum CPU seconds per process</td>
<td>Unlimited</td>
</tr>
<tr>
<td>(limit name)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cputime (time)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>filesize</td>
<td>Maximum</td>
<td>Unlimited, or as large as the file system allows.</td>
</tr>
<tr>
<td>Setting Name</td>
<td>Description</td>
<td>Recommended Value</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ulimit name</td>
<td>size for a</td>
<td>Your system</td>
</tr>
<tr>
<td>(limit name)</td>
<td>single file</td>
<td>administrator may</td>
</tr>
<tr>
<td>(file)</td>
<td></td>
<td>enforce limits on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the maximum size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of files for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reasons unrelated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to MicroStrategy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value must</td>
</tr>
<tr>
<td></td>
<td></td>
<td>be at least as</td>
</tr>
<tr>
<td></td>
<td></td>
<td>large as the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>maximum size for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>core dump files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(coredumpsize).</td>
</tr>
<tr>
<td>datasize</td>
<td>Maximum</td>
<td>Unlimited, or as</td>
</tr>
<tr>
<td>(data)</td>
<td>heap size</td>
<td>large as the</td>
</tr>
<tr>
<td></td>
<td>per process</td>
<td>system virtual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>memory allows.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Your system’s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>virtual memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>constraints</td>
</tr>
<tr>
<td></td>
<td></td>
<td>affect the data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>size you can set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for a process's</td>
</tr>
<tr>
<td></td>
<td></td>
<td>heap size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The value should</td>
</tr>
<tr>
<td></td>
<td></td>
<td>be the same as</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the maximum size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for core dump</td>
</tr>
<tr>
<td></td>
<td></td>
<td>files (coredumpsize).</td>
</tr>
<tr>
<td>stacksize</td>
<td>Maximum</td>
<td>200 MB</td>
</tr>
<tr>
<td>(stack)</td>
<td>stack size</td>
<td></td>
</tr>
<tr>
<td></td>
<td>per process</td>
<td></td>
</tr>
<tr>
<td>coredumpsize</td>
<td>Maximum size</td>
<td>Set this value to</td>
</tr>
<tr>
<td>(coredump)</td>
<td>for a single</td>
<td>the same value as</td>
</tr>
<tr>
<td></td>
<td>core dump</td>
<td>the maximum heap</td>
</tr>
<tr>
<td></td>
<td>file</td>
<td>size per process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(datasize). If core</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dump files are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>created that are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>larger than this</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value, the files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>are corrupted and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>unusable.</td>
</tr>
<tr>
<td>memoryuse</td>
<td>Maximum size</td>
<td>Unlimited, or as</td>
</tr>
<tr>
<td>(memory)</td>
<td>of physical</td>
<td>large as the</td>
</tr>
<tr>
<td></td>
<td>memory</td>
<td>physical memory of</td>
</tr>
<tr>
<td></td>
<td>allotted per</td>
<td>your system allows</td>
</tr>
<tr>
<td></td>
<td>process</td>
<td></td>
</tr>
<tr>
<td>vmemoryuse</td>
<td>Maximum size</td>
<td>Unlimited, or as</td>
</tr>
<tr>
<td>(vmemory)</td>
<td>of virtual</td>
<td>large as your</td>
</tr>
<tr>
<td></td>
<td>memory</td>
<td>system virtual</td>
</tr>
<tr>
<td></td>
<td>allowed per</td>
<td>memory allows</td>
</tr>
<tr>
<td></td>
<td>process</td>
<td></td>
</tr>
</tbody>
</table>
Methods of Installation

The methods of MicroStrategy installation are:

Command Line

In command line mode, you type the appropriate information at the prompt and press Enter. Instructions are included on each page of the MicroStrategy Installation Wizard.

In some cases, you are asked to make a selection by pressing 1 or 2, and Enter. You then press 0 and Enter to continue.

Defaults appear next to each prompt and are enclosed in square brackets, for example, [1]. Click Enter to use the default, or type a different response to the prompt to override the default.
In addition, on the command line wizard pages, the following options are available:

- Press 1 and then click **Enter** to proceed to the next page.
- Press 2 and then click **Enter** to return to the previous page.
- Press 3 and then click **Enter** to cancel the installation and close the MicroStrategy Installation Wizard.
- On the last page, which is MicroStrategy Installation Wizard Complete, press 3 and then click **Enter** to complete the setup and close the wizard.

For information on command line installation, refer to *Chapter 3, Installing MicroStrategy on Linux*.

**Silent Installation**

A silent, or unattended, installation is one that presents no graphical user interface (GUI). Silent installations are useful for system administrators who do not want users to run the installation themselves. It allows you to automate the installation, so it can be called from a script and executed without user interaction.

For information on silent installation, refer to *Silent Installation, page 616* in *Chapter 9, Automated Installation on Windows* and *Silent Installation, page 625* in *Chapter 10, Automated Installation on Linux*.

**Creating Custom Installation Packages**

You can reduce the amount of data that has to be downloaded for an installation by providing only the files required to complete a MicroStrategy installation. This technique can then be used to reduce the amount of data packaged and downloaded for other MicroStrategy installations within your organization.

The steps below show you how to create these custom installation packages.
If you are performing a MicroStrategy Update 1 installation, you must include all of the files provided as part of the Update installation in their default location. This means that you cannot use the steps below to create a custom MicroStrategy Update 1 installation package.

To Create a Custom MicroStrategy Installation Package

1. Retrieve the MicroStrategy installation files from the installation disk or the MicroStrategy download site. Save these files to a folder. Contact your MicroStrategy sales representative to determine the location and login credentials for the MicroStrategy download site.

2. Within the location where you saved the MicroStrategy installation files, browse to the **DataFiles** folder.

3. You can determine the required installation files in the following ways:

   - For Windows installations, you can use the MicroStrategy Installation Wizard to determine the required files, as described in *Move all of the compressed .zip files within this folder to a different folder location.*, page 109.

   - For Linux installations, the table below lists which installation files are required for each MicroStrategy component. Once you determine the required installation files, you can include them in your custom installation as described in *Creating Custom Installation Packages*, page 106 below.

<table>
<thead>
<tr>
<th>Installation File</th>
<th>MicroStrategy Components That Require The Installation File</th>
</tr>
</thead>
<tbody>
<tr>
<td>mstr1.tzp</td>
<td>All MicroStrategy components and products</td>
</tr>
<tr>
<td><strong>Installation File</strong></td>
<td><strong>MicroStrategy Components That Require The Installation File</strong></td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>mstr3.tzp</td>
<td>MicroStrategy Intelligence Server and all of its components</td>
</tr>
<tr>
<td>mstr4.tzp</td>
<td>MicroStrategy Web, including Web Analyst, Web Reporter, and Web Professional</td>
</tr>
<tr>
<td>mstr5.tzp</td>
<td>MicroStrategy Web Services for Office</td>
</tr>
<tr>
<td>mstr6.tzp</td>
<td>MicroStrategy Command Manager</td>
</tr>
<tr>
<td>mstr7.tzp</td>
<td>MicroStrategy Integrity Manager</td>
</tr>
<tr>
<td>mstr8.tzp</td>
<td>MicroStrategy System Manager</td>
</tr>
<tr>
<td>mstr9.tzp</td>
<td>MicroStrategy Mobile Client</td>
</tr>
<tr>
<td>mstr10.tzp</td>
<td>MicroStrategy Mobile Server</td>
</tr>
<tr>
<td>mstr11.tzp</td>
<td>MicroStrategy Portlets, which is a component of MicroStrategy Web</td>
</tr>
<tr>
<td>mstr12.tzp</td>
<td>MicroStrategy GIS Connectors, which is a component of MicroStrategy Web</td>
</tr>
<tr>
<td>mstr13.tzp</td>
<td>All MicroStrategy components and products</td>
</tr>
<tr>
<td>mstr14.tzp</td>
<td>MicroStrategy Enterprise Manager</td>
</tr>
<tr>
<td>mstr15.tzp</td>
<td>MicroStrategy MicroStrategy Identity Server</td>
</tr>
<tr>
<td>mstr16.tzp</td>
<td>MicroStrategy MicroStrategy Identity Manager</td>
</tr>
<tr>
<td>mstr18.tzp</td>
<td>MicroStrategy MicroStrategy Communicator</td>
</tr>
<tr>
<td>mstr20.tzp</td>
<td>MicroStrategy Messaging Services</td>
</tr>
<tr>
<td>mstr21.tzp</td>
<td>MicroStrategy Library Web &amp; Mobile</td>
</tr>
<tr>
<td>mstr22.tzp</td>
<td>MicroStrategy Collaboration Server</td>
</tr>
</tbody>
</table>
### Installation and Configuration Guide

<table>
<thead>
<tr>
<th>Installation File</th>
<th>MicroStrategy Components That Require The Installation File</th>
</tr>
</thead>
<tbody>
<tr>
<td>mstr25.tzp</td>
<td>Node.js which is installed with Intelligence Server and Collaboration Server</td>
</tr>
<tr>
<td>mstr26.tzp</td>
<td>Application Schema Server (Early Adopter)</td>
</tr>
<tr>
<td>mstr27.tzp</td>
<td>Redis which is installed with MicroStrategy Platform Analytics</td>
</tr>
<tr>
<td>mstr28.tzp</td>
<td>MicroStrategy Platform Analytics</td>
</tr>
<tr>
<td>mstr29.tzp</td>
<td>Common Files</td>
</tr>
<tr>
<td>mstr30.tzp</td>
<td>Community Connectors</td>
</tr>
<tr>
<td>mstr31.tzp</td>
<td>Certificate Manager</td>
</tr>
</tbody>
</table>

4. Move all of the compressed .zip files within this folder to a different folder location.

5. Use the MicroStrategy Installation Wizard to begin a MicroStrategy installation. For steps to locate and use the MicroStrategy Installation Wizard, see *Chapter 2, Installing MicroStrategy on Windows*.

6. Complete the steps up to and including the step to select the MicroStrategy components to be installed.

7. After selecting the MicroStrategy components to be installed and clicking **Next**, a message is displayed that lists the required installation files. Store all of these files in a location that can be accessed by the machine that will use the custom installation package.

8. Click **Cancel**.
Related Topics

Installation Files, page 578
Install on Demand Options, page 677

Licensing Information

If you have installed the Evaluation version of MicroStrategy, you cannot use its license key with a Generally Available (GA) license key in the same environment. Hence, the Evaluation version of MicroStrategy cannot be used for your production environment.

Types of Licenses

Refer to your MicroStrategy contract and any accompanying contract documentation for descriptions of the different MicroStrategy license types.

If you receive access to MicroStrategy Identity functionality as part of MicroStrategy analytics software products, either by purchasing licenses to such software products or receiving such software products as part of a maintenance upgrade, your use of this product is restricted to use solely for the purpose of authentication in conjunction with MicroStrategy analytics software products.

Installation and Configuration Checklists

This guide provides information on how to install and configure MicroStrategy products on Windows and Linux. To help you navigate through this guide, the following sections in this chapter list the chapters that you should refer to depending on the platform on which you are installing MicroStrategy products. Each list also provides a brief overview of each chapter. It is recommended that you read this
section before performing an installation. You can use the tables as checklists of installation and configuration tasks to be completed.

The appendixes in this guide are not listed in the checklists. The checklists only cover the main steps to install and configure MicroStrategy products. The appendixes in this guide contain important configuration details that are useful throughout the life cycle of your MicroStrategy installation.

Installing and Configuring MicroStrategy on Windows

If you are installing MicroStrategy on Windows, you should refer to the following sections sequentially.

- **Chapter 1, Planning Your Installation**: Review this chapter for important installation prerequisites and considerations.

- **Chapter 2, Installing MicroStrategy on Windows**: This chapter describes the procedures for installing the MicroStrategy products necessary to run your business intelligence application in a Windows environment.

Or

- **Chapter 9, Automated Installation on Windows**: As an alternative to the regular installation, you can perform a fully automated and unattended installation including customization routines available with the product. This chapter describes different types of unattended and automated installations and provides steps to perform these installations on Windows.

Additionally, **Chapter 11, Deploying OEM Applications** explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.

- **Chapter 5, Activating Your Installation**: After installing MicroStrategy products, you have 30 days to activate your software
installation. If you have not activated your software after 30 days, some MicroStrategy features may become unavailable until you complete the software activation.

- **Chapter 6, Configuring and Connecting Intelligence Server**: After installing and activating MicroStrategy products, you must use the MicroStrategy Configuration Wizard to configure the MicroStrategy metadata repository, statistics tables, history list tables, Intelligence Server, and project sources. This chapter describes the steps used to configure an installed MicroStrategy suite of products using the MicroStrategy Configuration Wizard.

- **Chapter 7, Deploying MicroStrategy Web and Mobile Server**: You can deploy your project to your user community using MicroStrategy Web. This chapter provides information on how to deploy and configure MicroStrategy Web on Windows and Linux platforms with various Web and application servers.

- **Chapter 13, Adding or Removing MicroStrategy Components**: This chapter describes the steps to add and remove MicroStrategy components on Windows, as well as other operating systems. For Windows platforms, refer to the following sections:
  - *Adding or Removing MicroStrategy Components on Windows, page 736.*

**Installing and Configuring MicroStrategy on Linux**

If you are installing MicroStrategy on Linux operating systems, you should refer to the following sections sequentially.

- **Chapter 1, Planning Your Installation**: Review this chapter for important installation prerequisites and considerations.
• Chapter 3, Installing MicroStrategy on Linux: This chapter describes the procedures for installing the MicroStrategy products necessary to run your business intelligence application on a Linux environment.

Or

• Chapter 10, Automated Installation on Linux: As an alternative, you can perform a fully automated and unattended installation without using the graphical user interface. This chapter describes different types of unattended and automated installations and steps to perform these installations on Linux.

Additionally, Chapter 11, Deploying OEM Applications explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.

• Chapter 5, Activating Your Installation: After installing MicroStrategy products, you have 30 days to activate your software installation. If you have not activated your software after these 30 days have passed, some MicroStrategy features may become unavailable until you complete the software activation.

• Chapter 6, Configuring and Connecting Intelligence Server: After installing and activating MicroStrategy products, you must use the MicroStrategy Configuration Wizard to configure the MicroStrategy metadata repository, statistics tables, history list tables, Intelligence Server, and project sources. This chapter addresses the processes necessary to configure an installed MicroStrategy suite of products using the Configuration Wizard. If no project sources are defined, then the Configuration Wizard opens.

Or

• Chapter 12, Configuring MicroStrategy Using Command Line Tools: MicroStrategy tools are provided in command line mode on Linux so
that you can perform various configuration tasks through the operating system console. This enables you to perform your required configurations even if you do not have access to the MicroStrategy interface.

- **Chapter 7, Deploying MicroStrategy Web and Mobile Server:** You can deploy your project to your user community using MicroStrategy Web. This chapter provides information on how to deploy and configure MicroStrategy Web and Web Universal on Windows and Linux platforms with various Web and application servers.

- **Chapter 8, Setting Up Documents and HTML Documents:** This chapter explains the setup required for the Intelligence Server to create and execute HTML documents and documents. It also describes the steps to create this setup, which are only necessary on a Linux environment.

- **Chapter 13, Adding or Removing MicroStrategy Components:** This chapter describes the steps to add and remove MicroStrategy components on all supported operating systems. For Linux platforms, refer to the following section:

  - Uninstalling MicroStrategy components on Linux
INSTALLING MICROSTRATEGY ON WINDOWS
This section describes the procedures for installing the MicroStrategy products that are necessary to run your business intelligence application on a Windows environment.

Before installing MicroStrategy products, see Chapter 1, Planning Your Installation for important pre-installation information.

Some MicroStrategy products are available in two versions, as described below.

- **Windows only**: The Windows only versions, labeled as MicroStrategy Intelligence Server, MicroStrategy Web (ASP.NET), and so on, are compatible only with a Windows platform. With these versions, MicroStrategy Web can be deployed quickly and easily using MicroStrategy's Internet Information Services (IIS) Web Server. The drawback is that IIS is the only Web server that can be used to deploy the Windows only version of MicroStrategy Web.

- **Universal (platform independent)**: The universal versions, labeled as MicroStrategy Intelligence Server, MicroStrategy Web (JSP), and so on, are compatible with Windows as well as Linux platforms. Installing the universal versions on Windows lets you deploy MicroStrategy Web with different application and Web server combinations. For example, instead of using IIS to deploy MicroStrategy Web, you can use Apache Tomcat, Oracle 10g, and so on.

If you have used the Evaluation Edition of the MicroStrategy platform, you may have installed most of these products already. However, additional considerations are important when you are setting up a production business intelligence system as opposed to running the evaluation software. You should read this chapter carefully, even if you already have a working system from your Evaluation Edition.

This section has the following information:

- **Installation Procedure in Windows**
Configuring your MicroStrategy Installation ........................................ 158

Additionally, Chapter 11, Deploying OEM Applications explains the
common workflow for deploying the MicroStrategy platform as an
Original Equipment Manufacturer (OEM) application.

Installation Procedure in Windows

The MicroStrategy Installation Wizard guides you through installing
one or more MicroStrategy products in a Windows environment. The
following sections can assist you in installing MicroStrategy products:

- Installing with the MicroStrategy Installation Wizard for Windows,
  page 118

- Installation Verification, page 143

There are installation alternatives and procedures to support your
MicroStrategy installation documented in this guide, including the
following:

- Prerequisites, see Installation Prerequisites, page 63 in Chapter 1,
  Planning Your Installation.

- Advanced installation functionality, such as installing in an SMS
  environment or using installation response files, see Chapter 9,
  Automated Installation on Windows.

- Installing and deploying MicroStrategy Web with other Web and
  application servers, see Chapter 7, Deploying MicroStrategy Web
  and Mobile Server.

- Deploying MicroStrategy Web Services ASP.NET and J2EE, see
  Enabling Single Sign-On with SAML Authentication in System
  Administration Guide.

- Prerequisites and procedures for MicroStrategy Identity, see
  MicroStrategy Identity Pre-Installation Instructions.
• If you have not uninstalled previous versions of MicroStrategy products, you are prompted to overwrite them. Click **Yes** to ensure that all products are installed properly. To retain the existing Tutorial metadata repository and warehouse, rename it or move it to another location before you start the installation process.

• Although MicroStrategy supports Windows Terminal Services, using Windows Terminal Services is not recommended for installation. It can affect the functionality of some MicroStrategy components.

**Installing with the MicroStrategy Installation Wizard for Windows**

To install MicroStrategy products, you must log on to your machine using a domain account with Windows administrative privileges for the domain or target machine. The domain must include your database servers.

To exit the installation process at any time, click **Cancel**.

---

**To access the MicroStrategy Installation Wizard**

1. Log on to the machine where you are installing one or more MicroStrategy products.

2. Exit all Windows applications before beginning the installation process.

3. Download the files from the MicroStrategy download site. Locate and run the **Setup.exe** file. Be aware of the following:

   • Contact your MicroStrategy sales representative to determine the location and login credentials for the MicroStrategy download site.
• You need to extract the downloaded files to locate the `Setup.exe` file. When extracting the files, ensure that the extraction software maintains the folder structure of the compressed files. Most extraction software maintains the folder structure by default, but if you use WinRAR, ensure that you select the Extract full paths option.

• To review an alternative, guided introduction to installing MicroStrategy software, you can locate and run the `MICROSTRATEGY.exe` file. For information on this installation alternative, see *Installing with a Guided MicroStrategy Introduction, page 144*.

• You can reduce the amount of data that has to be downloaded for the installation by excluding all of the `.zip` files, located in the `Installations/DataFiles` folder, from the download. You can use this technique to download only the files required to complete your MicroStrategy installation, which can then also be used to reduce the amount of data packaged and downloaded for other MicroStrategy installations. For steps to create these custom installation packages, see *Creating Custom Installation Packages, page 106*. Details on using a `response.ini` file to provide the location of the installation files are provided in *Chapter 9, Automated Installation on Windows* and the parameters used to specify the location of the required installation files are described in *Installation Files, page 578*.

4. If this is the first time you have installed MicroStrategy, you are prompted to choose the language for the wizard. Select the appropriate language from the drop-down list and click **OK**.

The MicroStrategy Installation Wizard opens and guides you through the rest of the installation process. The sections below describe the actions you must take for each page in the wizard. After you enter all
required information on an installation page, click **Next** to proceed to the next page of the installation routine.

If any services are running for previously installed MicroStrategy products, you are prompted to stop them. Click **Yes** to proceed. If you click **No**, you cannot install MicroStrategy products until you stop all MicroStrategy services.

**Welcome**

If you opened the MicroStrategy Installation Wizard through the Microsoft Control Panel using the Add/Remove Programs option, the wizard opens the Welcome page in maintenance mode. For more information on modifying, repairing, or removing all or part of your MicroStrategy installation, see *Chapter 13, Adding or Removing MicroStrategy Components*.

**License Agreement**

Read the license agreement, and accept or decline the agreement by clicking the appropriate button. If you decline, you cannot install MicroStrategy products.


**Customer Information**

Enter the following customer information:

- **User**
- **Company**
- **License Key**
To request a license key, go to the license key generator in the MicroStrategy Download Site, contact your MicroStrategy Representative or contact MicroStrategy Technical Support.

Install Options

Select one of the following install options:

- To install the entire platform on a single node environment, click Express. After installing the complete platform, you will have MicroStrategy Analytics, Mobility, and Security installed on your Windows server, as well as the required third-party software libraries.

  For steps to use the Express installation, see Performing a MicroStrategy Express installation.

- To install on an environment with multiple servers, or to select which MicroStrategy products to install, click Custom, and continue with the Installation Wizard.

Choose Destination Location

Browse to the locations where the MicroStrategy products and MicroStrategy common files are to be installed:

- **MicroStrategy Destination Folder**: Browse to and select the location where MicroStrategy products are installed. This is where executable files and other support files are installed for your licensed MicroStrategy products.

  While this setting determines the default root directory for the MicroStrategy products you install, you can change the destination of an individual product later as part of selecting which MicroStrategy products to install.
You can choose the directory for a product only if that product is not already installed on the server machine. Otherwise, the product can only be installed in the same directory in which it already exists.

- **MicroStrategy Common Files Destination Folder**: Browse to and select the location where MicroStrategy common files are installed. These files are required to support a MicroStrategy installation.

**Select Features**

Select the check box of a MicroStrategy product to include that product in the installation. Alternatively, you can clear a check box to uninstall or exclude a MicroStrategy product from the installation.

The installation pages you see after this step depend on the products you choose to install. These instructions describe all possible pages.

- If you are installing MicroStrategy Communicator, components of MicroStrategy Intelligent Enterprise are also installed.

  If you previously installed Community Connectors in 11.0 and are installing a newer version of MicroStrategy, your whitelist.txt file will be overwritten.

  To use your 11.0 whitelist.txt file in a newer version of MicroStrategy, backup the whitelist.txt file located in
  
  `<installPath>\Tomcat\CommunityConnectors\WEB-INF`

  before performing the upgrade install. After the installation, replace the new whitelist.txt with the backed up whitelist.txt file from 11.0.

Many of the platform components have subcomponents. If you expand the different MicroStrategy products, you can select the appropriate check boxes to specify the subcomponents to install. For information on MicroStrategy components and subcomponents, see *MicroStrategy Products and Components, page 11* in *Chapter 1, Planning Your Installation*.
You can see only MicroStrategy products that are available with your license key.

Destination Folder

You can select MicroStrategy products and their subcomponents to define their installation locations. When you select a MicroStrategy product or subcomponent, the Destination Folder area near the bottom of the interface displays the current installation folder for the product. Click **Browse** to select a different installation folder.

If you select a MicroStrategy product or subcomponent and the Browse button is not accessible, this means that the installation location cannot be changed. For example, if you select MicroStrategy Mobile you cannot define an installation location. However, if you expand this product, you can define the installation location for its subcomponents.

MicroStrategy Setup and Choose Data Files Location

You see the MicroStrategy Setup dialog box and the Choose data files location page only if some of the files, required to install the MicroStrategy components you have selected for installation, are not available. If you are using this technique to reduce the amount of data that has to be downloaded for the installation, it is recommended that you do the following:

- Review the files listed in the MicroStrategy Setup dialog box, and make a note of all the required files. These files need to be provided as part of the installation for the MicroStrategy components you selected using the Select Features page of the installation (see *Select Features, page 122*).

- Provide the location of the installation files using a **response.ini** file. This lets you access the installation files stored on a folder or
stored at a URL and accessed using HTTP or HTTPS. Details on using a response.ini file as part of an installation are provided in Chapter 9, Automated Installation on Windows and the parameters used to specify the location of the required installation files are described in Installation Files, page 578.

If the files required for the installation are stored in a folder, you can instead click Change on this Choose data files location page to navigate to and select the folder that stores the installation files. If all the required installation files are provided in the folder you select, you can click Next to continue the MicroStrategy installation.

Topology Configuration

Topology allows administrators to monitor MicroStrategy Services and manage them in MicroStrategy Workstation. The following functionality is supported:

- **Service Health Monitor**
  The administrator can monitor MicroStrategy services through Workstation and visualize if they are running or stopped.

- **Service Start/Stop**
  The administrator can start or stop MicroStrategy services through Workstation. This functionality is currently limited to SSH authentication with username and password.

See the topic, *How to View Environment Topology and Monitor Services* in the Workstation Online Help for more information about using topology within Workstation.

The two following types of lightweight agents are used to monitor services in Workstation:
• **Monitoring Agents**

The monitoring agents come with the installation of most MicroStrategy services. Use them to monitor services and view their health status.

• **Communication Agents:**

Some monitoring agents also act as communication agents. In addition to monitoring services, communication agents help other monitoring agents locate each other and gather monitoring information. The administrator must pick at least one agent to act as a communication agent, so all other agents can locate each other through this agent.

On environments where three or more machines host MicroStrategy Services, it is recommended that you choose at least three communication agents to provide redundancy and improved reliability, in case one communication agent becomes unavailable. The machines chosen to act as communication agents must be machines that host MicroStrategy Services, such as Intelligence Server or MicroStrategy Library. If you decide to use only one communication agent, it is recommended that you choose the machine that houses MicroStrategy Library (if deployed through the MicroStrategy installation), since Library communicates directly with Workstation.

If your Library deployment is done through a WAR file or if you do not use a machine that houses MicroStrategy Library to host the communication agent, then use a machine that houses Intelligence Server.

MicroStrategy uses Consul technology for Services Registration. A monitoring agent corresponds to a Consul agent in client mode. A communication agent corresponds to a Consul agent in server mode.
1. On the Topology Configuration dialog of the Installation wizard, select whether your environment has **Single** or **Multiple** machines.

2. If your environment contains multiple machines, determine which machines in your environment you want to act as communication agents. Among all the machines in your environment that will host MicroStrategy Services, select at least one of them (three are recommended for multiple machine environments) to act as a communication agent. Enter each machine in the text field, separated by semicolons. You must input the exact same list on all machines, using the Topology Configuration dialog of the Installation wizard, including the monitoring and communication agent machines.

The MicroStrategy Services are as follows:

- MicroStrategy Intelligence Server
- MicroStrategy Web Universal
- MicroStrategy Library
- MicroStrategy Mobile
- MicroStrategy Messaging Services
- MicroStrategy Platform Analytics
- MicroStrategy Certificate Store
- MicroStrategy Identity

An odd number of communication agents is required due to the leadership selection algorithm.

Examples:

servername1.domain.com;servername2.domain.com;servername3.domain.com;servername1.domain.com;

Example: Your environment contains the following two machines and you select one machine to host the communication agent:

- Machine 1 hosts MicroStrategy Library and the communication agent
- Machine 2 hosts the Intelligence Server

When performing the installation, enter the full domain name, **machine1.domain.com**, on both machines.

Example: Your environment contains the following three machines and you want all machines to host a communication agent:

- Machine 1 hosts MicroStrategy Library
- Machine 2 hosts Intelligence Server 1
- Machine 3 hosts Intelligence Server 2

When performing the installation, enter **machine1.domain.com;machine2.domain.com;machine3.domain.com**; on all three machines.
Example: Your environment contains the following five machines and you select three machines to host the communication agents:

- Machine 1 hosts MicroStrategy Library
- Machine 2 hosts Intelligence Server 1 and a communication agent
- Machine 3 hosts Intelligence Server 2 and a communication agent
- Machine 4 hosts Intelligence Server 3 and a communication agent
- Machine 5 hosts Intelligence Server or other services

Enter `machine1.domain.com;machine2.domain.com;machine3.domain.com;` on all machines.

Some important information to keep in mind:

- In MicroStrategy 2019, OpenSSH is no longer installed on Windows machines. You must install Open SSH to start and stop services using topology. Upgrading your system from 11.0 to 2019 removes Open SSH.

If consul is already installed on a machine, prior to installing MicroStrategy products, the service registration auto configuration and execution is affected. We do not recommend a separate consul installation on the same machine.

- If you select **Multiple machine environment**, you must enter an odd number of machines.

- If your environment includes more than one machine and machines with dynamic IP addresses as communication agents, use FQDN for the communication agent machine list during the installation.
If your environment includes more than one machine and machines with more than one IP address as communication agents, it is recommended that you use FQDN for the communication agent machine list during the installation. If IP address list is used, make sure the machine with multiple IP addresses can be pinged with an IP address included in the list.

Topology supports installation on machines with public IP addresses. To avoid unexpected joins to topology nodes, possibly exposing service information, you must configure your firewall correctly. Your firewall should allow ports for Server RPC (default 8300), and Serf LAN (default 8301), and Serf WAN (default 8302) only for the nodes within the Workstation topology node cluster. See Enabling Topology Communication Through a Firewall for more information.

After installing the topology feature, see Installing and Configuring Topology for additional instructions on configuring this feature to work in your environment.

MicroStrategy Identity Configuration

You see this page if you have selected to install MicroStrategy Identity. If you do not have all the information and want to configure this product later, click Skip to proceed with the installation.

- **SSL Certificate Authority Certificate**: The file that contains the trusted Root CA, Intermediate Root CA bundle (.pem). It must be the complete certificate chain for your SSL Server Certificate that you obtained from your IT Administrator.

- **SSL Server Certificate**: The server certificate (.crt) file for your Windows server.

- **SSL Server Certificate Key**: The key for your SSL server certificate (.key) file.
- **SSL Certificate Authority Key File Password**: If your CA-signed certificate has a password, create a text file containing this password and enter the text file location.

  To ensure that the SSL certificates are valid for your installation, you can run the following checks:

  1. The results for the following two commands should be identical.

     - openssl.exe x509 -noout -modulus -in sample.crt | openssl.exe md5
     - openssl.exe rsa -noout -modulus -in sample.key | openssl.exe md5

  2. No error is produced when the command openssl.exe verify -partial_chain -CAfile sample.pem sample.crt is executed.

- **SMTP Server**: Your company's SMTP server, followed by the port number in the next box.

- **SMTP Authentication**: If your server is password protected, then enter the username and password for the server.

- **Email Sender Address**: The email address that is authorized to send emails from your SMTP server, and will be used to send badge invitations for your MicroStrategy Identity network.

- **Host Name**: Enter the Fully Qualified Domain Name you are using, for example, yourFQDN.com.

**Download Open Source Software**

The Download Open Source Software dialog is displayed for informing users to download MySQL and related drivers, which are required for bundled MySQL and Platform Analytics database configuration.
Platform Analytics Data Warehouse

If Platform Analytics is chosen for installation you need to identify the MySQL database to store the Platform Analytics telemetry data.

Select the checkbox to define your own dedicated MySQL database or the bundled MySQL installation will be used as default.
For post-installation configuration instructions, see the Platform Analytics help.

Messaging Services Cluster Configuration

If Intelligence Server or Platform Analytics are chosen for installation, the option to configure a Messaging Services cluster is presented.

If no cluster needs to be configured, click Next.

To configure a cluster:

1. Select the I need to create a Messaging Services cluster for Platform Analytics check box.

2. Local Node: Provide the address for the local node.

3. Remote Nodes: Provide a comma separated list of hostnames for remote nodes in the cluster. The node id is determined by alphabetical order in node list.

For detailed configuration information see, MicroStrategy Messaging Services
Server Activation

If you have installed one or more MicroStrategy server products, you can request an Activation Code to activate your MicroStrategy server products upon completion of the installation process. The next few pages of the installation process guide you in providing the information you must submit to MicroStrategy to request an Activation Code. MicroStrategy server products include:

- MicroStrategy Intelligence Server
- MicroStrategy Web
- MicroStrategy Mobile Server

This page includes a welcome statement for the software activation process. Read the welcome statement and click Next to proceed.

Server Information

Specify information about your MicroStrategy server installation. Enter the following characteristics:

- **Name**: Distinguishes the name of this MicroStrategy server product installation from any other MicroStrategy server product installations in your company.

- **Location**: Physical location of the machine on which MicroStrategy server products are installed.

- **Use**: Description of how the server is used.

Click Privacy Statement to view the MicroStrategy Privacy Statement.
Installer Information

Specify contact information of the person installing the software. After your installation is complete an email containing the Activation Code is sent to the email address you confirm in this software activation step. Enter the following installer information:

- Specify whether you are an employee of the licensed company or installing on behalf of the licensed company.
- Enter the necessary data into all text fields. Make sure the email address you enter is correct. This email address is the recipient of the Activation Code.

- Select the check box at the bottom of the page to receive notifications about product updates, events, and special offers from MicroStrategy.
- Click Privacy Statement to view the MicroStrategy Privacy Statement.

Contact Information

You see this page if you indicated that you are not an employee of the company licensed to use this software, and are installing the software on behalf of that company.

Specify contact information for the employee license to use the software. Enter the necessary data into all text fields. Make sure the email address you enter is correct. After your installation is complete an email containing the Activation Code is sent to the email address you confirm in this software activation step.

- Select the check box at the bottom of the page to receive notifications about product updates, events, and special offers from MicroStrategy.
- Click Privacy Statement to view the MicroStrategy Privacy Statement.
Request Activation Code

This page includes options to request an Activation Code now or at a later time. This page provides the following options:

- Select **Yes, I want to request an Activation Code** and click **Next** to request an Activation Code. The Activation Code is sent to the email addresses supplied in the Installer Information and Contact Information pages.

- Select **No, I will request the Activation Code at a later time** and click **Next** to request an Activation Code at a later time.

  If you choose to request an Activation Code at a later time, a message is displayed that instructs you how to request an Activation Code after the installation procedure is completed. For more instructions on requesting an Activation Code at a later time, see *Request an Activation Code, page 341 in Chapter 5, Activating Your Installation*.

You have a grace period of 30 calendar days to activate your installation. If you do not complete the activation before the grace period expires, your MicroStrategy product stops functioning until you activate it. If you wait to activate your installation, you receive periodic reminders.

Once you request an Activation Code, an email is sent to the email addresses you specify in the Installer Information and Contact Information pages of the software activation procedure. The email provides instructions on how to use the requested Activation Code to activate your software. To activate your installation, you can also use the steps given in *Activate Your Installation, page 344 in Chapter 5, Activating Your Installation*. 
CPU License Information

MicroStrategy Web (ASP.NET) Setting

You see this page only if you choose to install MicroStrategy Web (ASP.NET) and only if you do not have a previous version of MicroStrategy Web installed.

Specify the Internet Information Services (IIS) virtual directory to be created for MicroStrategy Web pages. The default is MicroStrategy. In IIS, a virtual directory is the home location for a set of Web pages that the Web server hosts.

- If you have a previous version of MicroStrategy Web installed on the machine, the new version you install uses the same virtual directory the previous version is using. Therefore, you are not prompted to specify the name of the virtual directory.

- The name provided for a virtual directory must be unique. You cannot use the same name as the default for other MicroStrategy products.

- MicroStrategy automatically configures the MicroStrategy Web virtual directory to run with the version of .NET Framework that it requires.

MicroStrategy Web (ASP.NET) CPU Affinity Setting

You see this page only if you choose to install MicroStrategy Web (ASP.NET) and if the MicroStrategy Web installation detects that the license key entered is a CPU-based license. This page is not displayed on single-processor machines.

Specify the number of CPUs that MicroStrategy Web is licensed to use on the machine. You can specify only the number of CPUs that are allowed by the license. If MicroStrategy Web is installed on more than one machine, the total number of CPUs should not exceed the
maximum number of CPUs specified by the license. For machines that support hyper threading technology, the CPU counts correspond to physical CPUs, not logical CPUs.

To allow the setting to take effect, the installation stops IIS. After IIS has been restarted, the MicroStrategy Web application uses the specified number of CPUs.

For more information on the MicroStrategy Web CPU affinity feature, refer to the System Administration Guide.

**MicroStrategy Mobile Server (ASP.NET) Setting**

You see this page only if you choose to install MicroStrategy Mobile Server (ASP.NET) and only if you do not have a previous version of MicroStrategy Mobile Server installed.

Specify the Internet Information Services (IIS) virtual directory to be created for MicroStrategy Mobile Server. The default is MicroStrategyMobile. The virtual directory is part of the URL used to access the interactive reporting and analysis applications deployed on this machine via Mobile Server.

Mobile Server can be deployed using the same techniques used to deploy MicroStrategy Web, as described in *Deploying MicroStrategy Web and Mobile Server, page 457*. For additional configurations required to deploy Mobile Server, see the MicroStrategy Mobile Administration Guide.

- If you have a previous version of MicroStrategy Mobile Server installed on the machine, the new version you install uses the same virtual directory the previous version is using. Therefore, you are not prompted to specify the name of the virtual directory.
• The name provided for a virtual directory must be unique. You cannot use the same name as the default for other MicroStrategy products.

• MicroStrategy automatically configures the MicroStrategy Mobile Server virtual directory to run with the version of .NET Framework that it requires.

MicroStrategy Subscription Portal Setting

You see this page only if you choose to install MicroStrategy Subscription Portal, which is a component of Narrowcast Server, and only if you do not have a previous version of Subscription Portal installed.

Specify the name of the IIS virtual directory to be created for MicroStrategy Subscription Portal pages. The default is NarrowcastServer. In IIS, a virtual directory is the home location for a set of Web pages that the Web server hosts.

Subscription Portal offers you the ability to subscribe to and view Narrowcast Server services, service descriptions, and their most recent modification dates on the Web. For complete information about Subscription Portal and other components of Narrowcast Server, refer to the MicroStrategy Narrowcast Server documentation.

The name provided for a virtual directory must be unique. You cannot use the same name as the default for other MicroStrategy products.

MicroStrategy MDX Cube Provider Setting

You see this page only if you choose to install the MicroStrategy MDX Cube Provider and if you do not have a previous version installed.

Specify the virtual directory to be created for the MicroStrategy MDX Cube Provider. The default is MicroStrategyMDX. This virtual
directory is used as part of the URL to connect to TM1 data sources or Microsoft Analysis Services data sources for integration with MicroStrategy. For information on connecting to these MDX cube data sources, see the MDX Cube Reporting Guide.

MicroStrategy Intelligence Server Setting

You see this page if you choose to install MicroStrategy Intelligence Server, and if you do not have a previous version of Intelligence Server installed.

Select the check box to use the local system account as the Intelligence Server service account. If you clear the check box to set a different Intelligence Server service account, enter the following information:

- **Login**: A Windows login of the form `Domain\User` with full administrative privileges under which to run the Intelligence Server service.

  The user account used to run Intelligence Server must have full administrator privileges for the local machine. If the administrator default privileges have been modified for the user account, connection errors can occur. For example, if the user account is denied access to the DSN accessed by Intelligence Server, Intelligence Server connection fails.

- **Password**: A valid password for the Windows login entered in the Login box

- **Confirmation**: Retype the password to confirm it is correct

  If the password you supply changes, you must reconfigure the Windows service to use the new password. Otherwise, Intelligence Server connections fail when the connection attempts to authenticate the login and password.
MicroStrategy Narrowcast Server Setting

You see this page if you choose to install MicroStrategy Narrowcast Server, and if you do not have a previous version of Narrowcast Server installed.

Select the check box to bypass the creation of a Narrowcast Server service account.

It is recommended you create the Narrowcast Server service account. Clear the check box, and enter the following information:

- **Login**: A Windows login of the form `Domain\User` with administrative privileges under which to run the Narrowcast Server service

- **Password**: A valid password for the Windows login entered in the Login box

- **Confirmation**: Retype the password to confirm that it is correct

If you change the password for this account, you must reconfigure the Narrowcast Server Windows services to use the new password.

Review Installation Settings

This screen provides a summary of the MicroStrategy products and services you have selected to install along with the destination folder where they will be saved. If you wish to automatically restart your computer when the installation is complete, select the check box below the list of products and services as shown below. Restarting your computer after installation will ensure that the system configuration is completed properly.
If you do not choose to restart automatically, you can restart manually when the Installation Wizard is complete.

MicroStrategy Installation Wizard Complete

Select **Yes I want to restart my computer now** as pictured below. Restarting now will ensure the installation process is completed successfully.

Click **Finish**.

If you encounter errors while installing MicroStrategy, refer to **Appendix B, Troubleshooting**.

Default Passwords for Configuration

During the configuration process, the MicroStrategy Installer randomly generates a password for the MySQL Database Server, the MicroStrategy Web Administration page, and the MicroStrategy Mobile Administration page. The username and password for all three are the same.
- **Username**: mstr

- **Password**: The password can be found in the following file on the server where MicroStrategy Secure Enterprise was installed:

  C:\Program Files (x86)\Common Files\MicroStrategy\express_password.txt

Note the username and password, and store them securely. These credentials will be used to configure the DSN to the Platform Analytics project.

It is recommended to delete the express_password.txt file after the password is stored securely. This password cannot be recovered if it is lost.

### Installation Verification

During the installation routine, the MicroStrategy Installation Wizard gathers and records information about your system and your installation selections. You can verify installation setup information through the installation log file (install.log), located by default in:

- **32-bit Windows environments**: C:\Program Files\Common Files\MicroStrategy.

- **64-bit Windows environments**: C:\Program Files (x86)\Common Files\MicroStrategy.

The installation log file includes the following information:

- Installation date
- Target directories
- Program folder name
- Operating system identification
- Hardware specifications
• Selected installation options
• Registry paths
• List of registered files

The installation log file can be helpful if you encounter errors during the installation process. For example, the log can tell you if a registry key or path was not added or if a critical file was not registered successfully.

Installing with a Guided MicroStrategy Introduction

The installation procedure provided in this chapter assumes that you use the Setup.exe file to install MicroStrategy software. As an alternative, a guided introduction to MicroStrategy software and the installation process is also provided. This introduction is provided as an Adobe Flash visualization.

To use this Flash visualization, you must locate and run the file MICROSTRATEGY.exe, which is available in the MicroStrategy install media or the files downloaded from the MicroStrategy download site. You can then use the Flash visualization to review documentation on MicroStrategy software, as well as begin the installation process.

If you provide the MicroStrategy installation files on a network location, you must map a network drive for users to access the MICROSTRATEGY.exe file. If users run this file without locating it through the use of a mapped network drive, the links to open various product manuals will not function properly.

To continue with the installation procedure, see Welcome, page 120.

Performing a MicroStrategy Express installation

The Express option installs your MicroStrategy Secure Enterprise Platform with all the features of Analytics, Mobility, and MicroStrategy
Identity. This guide describes the Express Install option, which installs the entire platform on a single machine.

The MicroStrategy Express option installation is specifically designed for deployments of up to 15-20 concurrent users with the recommended hardware specifications. This makes it an ideal solution for a development environment, or for evaluating the MicroStrategy Secure Enterprise Platform capabilities.

You need to provide some information for the Analytics and MicroStrategy Identity configuration.

After installing the complete Platform, you will have MicroStrategy Analytics, Mobility, and MicroStrategy Identity installed on your Windows server, as well as the required Third-party software libraries. For a complete list of all MicroStrategy components, see What you are installing.

Prerequisites

- System requirements:
  

  - Windows 7 (64-bit) and Windows 10 (64-bit) are supported for demo purposes.


  - All components are installed on the local C:\ drive, which requires 12 GB of disk space.

  - To successfully complete the installation process, your server must not have any MicroStrategy components installed.

  - 8 GB of RAM
• Multi-core 64bit processor

• MicroStrategy Badge successfully installed on your iPhone or Android phone.

• MicroStrategy Secure Enterprise software requirements:

  • Download and extract the MicroStrategy installation package from the MicroStrategy Download Site at https://download.microstrategy.com/. In the extracted files, locate MicroStrategy.exe or Setup.exe.

  • Your MicroStrategy software license key is for 64-bit servers. To request a license key, go to the license key generator in the MicroStrategy Download Site at https://software.microstrategy.com, contact your MicroStrategy Representative or contact MicroStrategy Technical Support at support@microstrategy.com.

  • After installation, a MicroStrategy Landing page containing links to the main MicroStrategy Platform services is displayed. The page requires JavaScript to be enabled to execute inside a web browser. Contact your IT administrator for assistance.

  • If Microsoft Internet Information Server (IIS) is present on the machine, the Express installation includes Subscription Portal and the MDX Cube Provider. If IIS is not present, these two components are not installed.

• Connectivity requirements:

  • Your Windows computer must be accessible through a Fully Qualified Domain Name.

  • Your mobile device must be able to connect to your Windows server. If not, you will not be able to retrieve your badges with
MicroStrategy Badge or download your dossiers from the MicroStrategy Mobile client.

- The MicroStrategy services listed in *Ports and connectivity information* must be able to communicate.

- The following Windows firewall inbound rules must be added:

<table>
<thead>
<tr>
<th>Name</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Tomcat 8</td>
<td>C:\Program Files (x86)\Common Files\MicroStrategy\Tomcat\apache-tomcat-8.0.30\bin\tomcat8.exe</td>
</tr>
<tr>
<td>Apache 2.4</td>
<td>C:\Program Files (x86)\Common Files\MicroStrategy\Apache\Apache24\bin\httpd.exe</td>
</tr>
<tr>
<td>MicroStrategy Intelligence Server x64</td>
<td>C:\Program Files (x86)\MicroStrategy\Intelligence Server\MSTRSvr2_64.exe</td>
</tr>
<tr>
<td>MicroStrategy Open Refine Server x64</td>
<td>C:\Program Files (x86)\MicroStrategy\Intelligence Server\MJRefSvr_64.exe</td>
</tr>
<tr>
<td>MySQL</td>
<td>C:\Program Files (x86)\Common Files\MicroStrategy\MySQL\mysql-5.6.28-winx64\bin\mysqld.exe</td>
</tr>
</tbody>
</table>

- **MicroStrategy Identity** requires an SSL web server. If you are not familiar with these elements, contact your IT administrator for assistance. The following items are needed to successfully configure and access MicroStrategy Identity:
  - An SSL Certificate Authority Root certificate and Intermediate certificates, appended into a chain (.pem) file. You need to obtain
this from a third-party signing authority, such as Verisign or Thawte. Contact your IT administrator for assistance.

- An HTTPS/SSL server certificate (.crt) file, and the matching key (.key) file. These must match the Fully Qualified Domain Name of your MicroStrategy Identity web services server.

- Certificate paths provided during installation must not be changed since the MicroStrategy Identity Server depends on the certificates to operate.

- To allow the MicroStrategy Identity Server to send invitation emails, provide the connectivity information for the email (SMTP) Server, including the port and credentials.
To start the Express installation

1. Log in to the Windows server as a user with administrator privileges.

2. In the installation folder, locate and run MicroStrategy.exe. Alternatively, you can locate and run Setup.exe. Accept the license agreement and continue.

3. On the Customer Information page, complete the following fields, which are used to create the administrator badge:
   - **First Name**: The first name of the administrator.
   - **Last Name**: The last name of the administrator.
   - **Email address**: The email address of the administrator. This is the email address where you will receive the badge invitation for your MicroStrategy Identity network.
   - **License Key**: The license key for your installation. This key is required to install the MicroStrategy Platform on a 64-bit Windows server. If you do not have this license key, contact your MicroStrategy Representative or support@microstrategy.com.

4. Click **Next**.

5. On the Install Options page, select **Express**, and click **Next**.

6. To configure MicroStrategy Identity, provide the following information. If you do not have this information, contact your IT Administrator.

   To use MicroStrategy Identity with MySQL 5.7.4 and above, the following line needs to be added or modified in MySQL my.cnf file, usually under /etc directory: `sql-mode="STRICT_`
CA Certificate chain: The file that contains the trusted Root CA, Intermediate Root CA bundle (.pem). It must be the complete certificate chain for your SSL Server Certificate that you obtained from your IT Administrator. The path must be specified in an absolute format such as C:\folder\example.pem.

SSL Server Certificate: The server certificate (.crt) file for your Windows server, specified using an absolute path.

SSL Server Certificate Key: The key for your SSL server certificate (.key) file, specified using an absolute path.

(Optional) SSL Certificate Key Password File: If your CA-signed certificate has a password, create a text file containing this password and enter the text file location, using an absolute path.

Email (SMTP Server) Server: Your company's SMTP server, followed by the port number in the next box.

Authentication (optional): If your server is password protected, then enter the username and password for the server.

Email Sender Address: The email address that is authorized to send emails from your SMTP server, and will be used to send badge invitations for your MicroStrategy Identity network.

Host Name: Type the Fully Qualified Domain Name of your Windows server, for example, webserver.acme-corporation.net.
If you do not have all the required information and want to manually configure MicroStrategy Identity later, click **Skip**. The subsequent steps assume you have entered all the information required to set up MicroStrategy Identity.

7. Click **Next**.

8. The MicroStrategy Express installation relies on MySQL open source components that are not provided by MicroStrategy. You can choose to provide the necessary MySQL components by following the instructions given in the installer or authorize the installer to download MySQL components on your behalf by clicking **Next**. You can review the terms of the GPL license v2.0 [here](#).

9. You can choose to add R and R Integration Pack to your installation to have these analytics tools installed and configured. Your options are:

   - **Option 1**: Click **Next** to have the files downloaded and installed for you.

   - **Option 2**: Click the **R** link to manually download the files to your `Downloads` folder to manually install them later. See the [R Integration Pack User Guide](#) for installation procedures.

   - **Option 3**: Click **Skip** to proceed without installing R analytics or R Integration Pack.

10. You have now entered all the information required to proceed. To change settings, click **Back**.

11. Check the box marked **Automatically restart my computer when the installation completes** if you wish to enable an automatic system reboot.
12. Click **Install**. Installation takes approximately 20 minutes with the recommended hardware.

13. If you did not choose to restart automatically, restart your machine now by clicking **Finish**. The MicroStrategy software is not configured until after a restart.

14. After your machine has restarted, log in with your administrative account. Platform configuration automatically begins. This can take up to 30 minutes with the recommended hardware.

   - On Windows 7, 2008, 2012, and 2012R2, it is normal to experience a black screen, for up to 15 minutes, before a progress bar is displayed.

   - On Windows 10, the operating system loads as normal, while the configuration continues in the background. Wait for the progress bar to display before resuming normal operations.

   - A progress bar displays while the post-configuration steps are processed.

   - You will receive two emails with instructions to retrieve your badges during this time period.

15. When the installation is complete, your default browser automatically opens the MicroStrategy Secure Enterprise launch page.

   - If the launch page does not display, use the shortcut on the desktop.

After the MicroStrategy installation is complete, you have 30 days to activate your installation. Before you activate your installation you must request an Activation Code from MicroStrategy. You can complete this request when you install MicroStrategy with the MicroStrategy Installation Wizard or after the installation using
MicroStrategy License Manager. For steps, see *Activating Your Installation*.

To log in to Identity Manager with MicroStrategy Badge

1. You should have received two badge invitation emails for your MicroStrategy Identity network:
   - **MicroStrategy Identity Network**: This badge can be used to change the configuration for your MicroStrategy Identity Server using Identity Manager.
   - **MicroStrategy Administrator**: This badge is used to log into Identity Manager and manage your MicroStrategy Secure Enterprise system (add users, connect to Active Directory, and so on).

     If you did not receive badge invitation emails or if the QR code does not load on the Identity Manager page, you may need to complete the MicroStrategy Identity Configuration. See To complete the MicroStrategy Identity Configuration (Windows and Linux).

2. From your mobile device:
   - Download MicroStrategy Badge for your iOS or Android device.
   - Open each email’s invitation link to individually download both badges for your MicroStrategy Identity network. Be sure that your mobile device is connected to the same network as your SMTP server.

3. On the landing page, click **Identity Manager**.

4. Open the **Scanner** tab in MicroStrategy Badge and scan the QR code. When prompted, select your MicroStrategy Identity Manager badge.
5. From Identity Manager, you can add users, and logical/physical resources.

To log in to MicroStrategy Tutorial

1. You can log into Tutorial by scanning a QR code with MicroStrategy Badge. Open the **Scanner** tab to scan the QR code.

   If you did not configure MicroStrategy Identity, you must log in using the MicroStrategy Administrator credentials, as described below:

   - Click **Credentials** below the QR code.
   - Type **Administrator** for the username and keep the password empty.
   - Click to log in.

2. After logging in, you can explore the capabilities of MicroStrategy by following the guided tutorials or exploring on your own, and creating your own dossiers.

To log in to the MicroStrategy Web Administration page, the MicroStrategy Mobile Administration page, or the MySQL database server

During the configuration process, the MicroStrategy Installer randomly generates a password for the MySQL Database Server, the MicroStrategy Web Administration page, and the MicroStrategy Mobile Administration page. The username and password for all three are the same.
• **Username**: mstr

• **Password**: The password can be found in the following file on the server where MicroStrategy Secure Enterprise was installed:

   C:\Program Files (x86)\Common Files\MicroStrategy\express_password.txt

Note the username and password, and store them securely. These credentials will be used to configure the DSN to the Platform Analytics project.

It is recommended to delete the `express_password.txt` file after the password is stored securely. This password cannot be recovered if it is lost.

### Supporting Information

What you are installing

**MicroStrategy Analytics and Mobility**

• MicroStrategy Intelligence Server

• MicroStrategy Web Server

• MicroStrategy Mobile Server

• MicroStrategy Collaboration Server

• MicroStrategy Narrowcast Server

• MicroStrategy Developer

• MicroStrategy Library

• MicroStrategy Messaging Services

• MicroStrategy Platform Analytics

• MicroStrategy Community Connectors
Third-party software libraries

The MicroStrategy installation installs third-party software libraries to provide a streamlined user experience. For a list of the libraries and versions, see the "System Requirements" in the Readme.

Ports and connectivity information

The MicroStrategy Express installation configures the following services on the specified ports. It also sets inbound Windows firewall rules to permit traffic on the specific ports. During an uninstall, the ports are closed on the Windows firewall.

- Tomcat: 8080
- Identity Manager: 443
- MicroStrategy Identity Server 1-way: 1443
- MicroStrategy Identity Server 2-way: 2443
- Identity Gateway: 9501
- MySQL: 3306
- Intelligence Server: 34952
- REST Server: 34962
- Collaboration Server: 3000
- Export Engine Micro-Service: 20100
- Apache ZooKeeper: 2181
- Apache Kafka: 9092
- Redis: 6379
- MongoDB: 27017

Certificate information

In addition to the certificates provided during installation, a Signing CA is generated to support all PKI signing operations of the MicroStrategy Identity Server, as well as a SAML Certificate used in all SAML transactions at the time of configuration. These certificates have an expiration of one year from the time of installation and are located in the **c:\program files (x86)\Common Files\MicroStrategy\Certificates** folder. The keys for the certificates can be found in the **c:\program files (x86)\Common Files\MicroStrategy\Keys** folder. To reconfigure this certificate, see the MicroStrategy Identity Server documentation.

If changes are made to the SSL certificates at a later date, the configuration files containing references to these certificates can be found in the following locations:

- **C:\Program Files (x86)\MicroStrategy\Usher\MicroStrategy Identity Server\usherApps\shardIDM\conf\server.xml**
In addition, Certificates from UsherCACertificate.pem are added to Java Key Store

UsherSAML.crt is saved to C:\Program Files (x86)\MicroStrategy\Usher\MicroStrategy Identity Server\ usherApps\shardIDM\webapps\files\saml2.crt

Configuring your MicroStrategy Installation

After completing the MicroStrategy Installation Wizard steps to install MicroStrategy products, you can set up and configure your
installation. To help guide the rest of your installation and configuration steps, refer to the section *Installing and Configuring MicroStrategy on Windows*, page 111 in *Chapter 1, Planning Your Installation*, for an installation and configuration checklist.

- The next section in the installation and configuration checklist and in this guide covers software activation steps with MicroStrategy. These steps should be done before or soon after the Configuration Wizard tasks mentioned below. For more information, refer to *Chapter 5, Activating Your Installation*.

- After restarting your machine to complete an initial MicroStrategy installation, the **MicroStrategy Configuration Wizard** opens. The Configuration Wizard allows you to configure your MicroStrategy production environment. For more information, refer to *Chapter 6, Configuring and Connecting Intelligence Server*. 
INSTALLING MICROSTRATEGY ON LINUX
This section describes the procedure for installing MicroStrategy on Linux platforms and covers the following:

Before installing MicroStrategy products, see Chapter 1, Planning Your Installation for important pre-installation information.

Additionally, Chapter 11, Deploying OEM Applications explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.

For supporting installation information, see Installing MicroStrategy on Windows.

Installation Procedures on Linux

The MicroStrategy products that you can install on Linux environments are:

- MicroStrategy Intelligence Server
- MicroStrategy Web
- MicroStrategy Portlets
- MicroStrategy GIS Connectors
- MicroStrategy Web Services for Office
- MicroStrategy Mobile
- MicroStrategy Command Manager
- MicroStrategy Integrity Manager
- MicroStrategy Messaging Services
- MicroStrategy System Manager
- MicroStrategy Identity:
  - MicroStrategy Identity Server
  - MicroStrategy Identity Manager
- MicroStrategy Communicator
- MicroStrategy Library
- MicroStrategy Platform Analytics

The MicroStrategy SDK and MicroStrategy Developer Library (MSDL) are not included in the MicroStrategy installation. You can download the MicroStrategy SDK from the MicroStrategy support site. You can also access the MicroStrategy Developer Library from the MicroStrategy support site.

R and R Integration Pack libraries are optional components that can be installed during MicroStrategy platform installation.

For more information about these products, see *MicroStrategy Products and Components, page 11*.

The following processes will be registered as OS services automatically following restart after installation is complete:

- Intelligence Server
- Enterprise Manager Service
- Data Wrangling process
- Hadoop Gateway Manager
- Kafka
- Zookeeper
- Intelligence Server Log consumer
- Export Engine
It is recommended that you install MicroStrategy products as the root user.

Install MicroStrategy MicroStrategy Identity, see *MicroStrategy Identity Pre-Installation Instructions.*

If you are installing MicroStrategy Identity Manager, you must create a database to manage user identities.

- If you are installing MicroStrategy products with a CPU-based license, you must be logged in as the root user; otherwise an error message is displayed and the installation fails.

- If you want a non-root user to be the administrator of the server, you must manually change the ownership after running the installation. Intelligence Server operation is dependent on root user privileges and permissions. Therefore, changing the ownership of Intelligence Server to a non-root user is not a certified or recommended practice.

- Only a user with root permissions can install MicroStrategy Identity components.

During installation, the user account for Intelligence Server is tested to verify that it can successfully support the use of common system tools for the operating system. If you change the user account for Intelligence Server, you must verify that this user account can use and access common system tools for the operating system.

- Script files within `HOME_PATH/env` and other configuration files within `HOME_PATH` (see *Installing with the MicroStrategy Installation Wizard for Linux* for information on this MicroStrategy directory) are overwritten anytime a new MicroStrategy product is installed on a machine. Backup copies of the previous file are also created during the installation. These backup copies can be used to update the new versions of the script and configuration files to include any prior modifications.
For example, Intelligence Server is installed on a machine. Then a week later Command Manager is installed on the same machine. During this installation of Command Manager, script files such as ODBC.sh are overwritten and a backup copy of each of these files is created before installing Command Manager.

Different methods of installation

MicroStrategy products can be installed on Linux, either in graphical user interface (GUI) mode or in command line mode, using the MicroStrategy Installation Wizard. In both cases, the MicroStrategy Installation Wizard runs, displaying the same pages and requesting the same information. The main differences are in how you provide the information and navigate through the wizard.

Using command line mode

In command line mode, you type the appropriate information at the prompt and press Enter. Instructions are included for each page of the MicroStrategy Installation Wizard.

In some cases, you are asked to make a selection by pressing 1 or 2, followed by pressing Enter. You then press 0 and Enter to continue.

Defaults appear next to each prompt and are enclosed in square brackets, for example, [1]. Click Enter to use the default, or type a different response to the prompt to override the default.

In addition, on the command line wizard pages, the following options are available:

- Press 1 and then click Enter to proceed to the next page.
- Press 2 and then click Enter to return to the previous page.
Press 3 and then click **Enter** to cancel the installation and close the MicroStrategy Installation Wizard.

On the last page, which is MicroStrategy Installation Wizard Complete, press 3 and then click **Enter**.

**Using silent mode**

You can perform a fully automated and unattended installation within the MicroStrategy platform when you do not have access to a Linux graphical user interface. This also lets you perform an installation on other machines.

For information on how to perform a silent installation on a Linux environment, see *Chapter 10, Automated Installation on Linux*.

**Installing with the MicroStrategy Installation Wizard for Linux**

To install MicroStrategy products, you must log on to your machine using a valid Linux account. For ease of management and maintenance, it is recommended that you create a dedicated user account.

You need root access permissions for installation if you have purchased the CPU-based MicroStrategy license.

If you want to enable additional error and troubleshooting issue logging for the MicroStrategy installation routine, contact MicroStrategy Technical Support.

To exit the installation process at any time, click **Cancel**.
To access the MicroStrategy Installation Wizard

1. Navigate to the MicroStrategy Installation folder and then QueryReportingAnalysis/Linux

You can access the installation files by asking your system administrator to share the files on a network location. There are different installation files for installing MicroStrategy products on different platforms; Windows and Linux.

You can reduce the amount of data that has to be downloaded for the installation by excluding all of the .tzp files located in the DataFiles folder. You can use this technique to download only the files required to complete your MicroStrategy installation, which can then also be used to reduce the amount of data packaged and downloaded for other MicroStrategy installations. For steps to create these custom installation packages, see Creating Custom Installation Packages, page 106. Details on using a options.txt file to provide the location of the installation files are provided in Chapter 10, Automated Installation on Linux and the parameters used to specify the location of the required installation files are described in Install on Demand Options, page 677.

2. Type one of the following commands, depending on the installation mode you chose:

- To run the wizard in GUI mode: ./setup.sh
- To run the wizard in command line mode: ./setup.sh -console
- To run the wizard in silent mode: ./setup.sh -silent -options options.txt
For information on performing a silent installation with an options.txt file, see Completing a Silent Installation, page 625.

3. The MicroStrategy Installation Wizard opens and leads you through the installation process. The following sections describe the actions you need to take for each page in the wizard.

To complete the installation, you must have write permissions in the installation directory; otherwise the installation fails.

After you enter all required information on an installation page, click Next, or press 1 and then click Enter, to proceed to the next page of the installation routine.

To quit the installation at any time during the setup, click Cancel, or press 3 and then click Enter.

The following dialogs appear in the Install wizard:

- Language Setup
- MicroStrategy Installation Selection
- License Agreement
- Customer Information
- Choose Destination Location
- Select Components
- Missing Installation Files
- Missing Requirements
- System Requirements
- MicroStrategy Identity Server Settings: Step 1
Linux Install Wizard - Language Setup

Specify the language to be used for the MicroStrategy installation.

Linux Install Wizard - MicroStrategy Installation Selection

This dialog appears if there are installations of MicroStrategy software on the current machine. The steps provided here assume that you are either installing for the first time or creating a new installation.

You can support multiple installations of MicroStrategy on Linux machines. Additionally, you can also modify, repair, and upgrade
existing MicroStrategy installations. This dialog provides the following installation options to support these scenarios:

- **Create a new installation**: Select this option to create a new installation of MicroStrategy on the machine. If an installation of MicroStrategy is already present on the machine, you can select this option to install a completely separate copy of MicroStrategy on the machine.

- **Use an existing installation**: If an installation of MicroStrategy is already present on the machine, you can select this option to perform various installation configurations. Select the installation you want to modify from the drop-down list, and then select one of the following installation configurations. The installation configuration options that are available depend on the type of installation that is being performed:
  
  - **Modify**: Select this option to add new program components or to remove currently installed components. If you want to remove all MicroStrategy components, use the Uninstall option described below. The remaining dialogs are the same as for a first-time installation, although some dialogs may be skipped if they are not required as part of the installation modification.
  
  - **Repair**: Select this option to re-install program components if you have problems with previously installed components. Your program components are returned to their original installation state. As part of a repair installation, you can also designate this machine as a Health Agent.

  - **Uninstall**: Select this option to uninstall all MicroStrategy components.

  - **Upgrade**: Select this option to upgrade all MicroStrategy components to the version you are installing. This option is only available if the version you are installing is a more recent version.
of MicroStrategy than the current installation. Only the MicroStrategy components currently installed are upgraded, you cannot install or uninstall MicroStrategy components as part of an upgrade. For best practices and steps to upgrade your MicroStrategy installation, see the Upgrade Guide.

- **Remove Update 1**: Select this option to uninstall a MicroStrategy Update 1 installation.

**Linux Install Wizard - License Agreement**

Read the license agreement and select to accept or decline the agreement. If you choose to decline, you cannot install MicroStrategy products.

**Linux Install Wizard - Customer Information**

Enter the following customer information:

- **User**
- **Company**
- **License Key**

To request a license key, go to the license key generator in the MicroStrategy Download Site, contact your MicroStrategy Representative or contact MicroStrategy Technical Support.

**Linux Install Wizard - Choose Destination Location**

Specify the locations where the MicroStrategy products and MicroStrategy common files are to be installed:
- **MicroStrategy Home Directory**: Specify the location where the MicroStrategy configuration files and application launchers are to be installed, according to the following guidelines:
  - The default location is /var/opt/MicroStrategy, or $HOME/MicroStrategy if you do not have write access to /var/opt/MicroStrategy.
  - Do not install the MicroStrategy configuration files directly to your Linux Home Directory ($HOME). To ensure that the required permissions can be defined for the MicroStrategy configuration files, you must install these files within a separate directory. For example, the default path of $HOME/MicroStrategy uses the MicroStrategy directory within $HOME to ensure permissions on these files are defined correctly.
  - The path specified for the home directory is referred to as HOME_PATH in this guide.
  - Do not change the names of folders within the HOME_PATH after installing Intelligence Server.
  - When including paths during a MicroStrategy installation, include absolute paths rather than relative paths.

- **MicroStrategy Install Directory**: Specify the location where the MicroStrategy products are to be installed, according to the following guidelines:
  - The default location is /opt/MicroStrategy, or $HOME/MicroStrategy/install if you do not have write access to /opt/MicroStrategy.
  - The path specified for the install directory is referred to as INSTALL_PATH in this guide.
• Do not change the names of folders within the INSTALL_PATH after installing Intelligence Server.

• When including paths during a MicroStrategy installation, include absolute paths rather than relative paths.

• MicroStrategy Log Directory: Specify the location where the MicroStrategy application logs are to be created, according to the following guidelines:

  • The default location is /var/log/MicroStrategy, or $HOME/MicroStrategy/log if you do not have write access to /var/log/MicroStrategy.

  • The path specified for the log directory is referred to as LOG_PATH in this guide.

  • When including paths during a MicroStrategy installation, include absolute paths rather than relative paths.

Linux Install Wizard - Select Components

Select a MicroStrategy product to include it in the installation. Alternatively, you can clear a checkbox to uninstall or exclude a MicroStrategy product from the installation.

The installation dialogs you see after this step depend on the products you choose to install. These instructions describe all possible dialogs.

If you are installing MicroStrategy Communicator, components of MicroStrategy Intelligent Enterprise are also installed.

If you previously installed Community Connectors in 11.0 and are installing a newer version of MicroStrategy, your whitelist.txt file will be overwritten.
To use your 11.0 whitelist.txt file in a newer version of MicroStrategy, backup the whitelist.txt file located in

\<installPath>\Tomcat\CommunityConnectors\WEB-INF

before performing the upgrade install. After the installation, replace the new whitelist.txt with the backed up whitelist.txt file from 11.0.

Many of the platform components have subcomponents. If you expand the different MicroStrategy products, you can select the appropriate checkboxes to specify the subcomponents to install. For information on MicroStrategy components and subcomponents, see *MicroStrategy Products and Components, page 11 in Chapter 1, Planning Your Installation*.

You can see only MicroStrategy products that are available with your license key.

**Destination Folder**

You can select MicroStrategy products and their subcomponents to define their installation locations. When you select a MicroStrategy product or subcomponent, the Destination Folder area near the bottom of the interface displays the current installation folder for the product. Click **Browse** to select a different installation folder.

If you select a MicroStrategy product or subcomponent and the Browse button is not accessible, this means that the installation location cannot be changed. For example, if you select MicroStrategy Mobile you cannot define an installation location. However, if you expand this product, you can define the installation location for its subcomponents.

**Linux Install Wizard - Missing Installation Files**

You see the Missing Installation Files message only if some of the files, required to install the MicroStrategy components you have selected for installation, are not available. If you are downloading
only a subset of the installation files to reduce the amount of data that has to be downloaded for the installation, it is recommended that you do the following:

- Determine the files required for the MicroStrategy components you are installing. A list of installation file requirements is provided in the table below:

<table>
<thead>
<tr>
<th>Installation File</th>
<th>MicroStrategy Components That Require The Installation File</th>
</tr>
</thead>
<tbody>
<tr>
<td>mstr1.tzp</td>
<td>All MicroStrategy components and products</td>
</tr>
<tr>
<td>mstr3.tzp</td>
<td>MicroStrategy Intelligence Server and all of its components</td>
</tr>
<tr>
<td>mstr4.tzp</td>
<td>MicroStrategy Web, including Web Analyst, Web Reporter, and Web Professional</td>
</tr>
<tr>
<td>mstr5.tzp</td>
<td>MicroStrategy Web Services for Office</td>
</tr>
<tr>
<td>mstr6.tzp</td>
<td>MicroStrategy Command Manager</td>
</tr>
<tr>
<td>mstr7.tzp</td>
<td>MicroStrategy Integrity Manager</td>
</tr>
<tr>
<td>mstr8.tzp</td>
<td>MicroStrategy System Manager</td>
</tr>
<tr>
<td>mstr9.tzp</td>
<td>MicroStrategy Mobile Client</td>
</tr>
<tr>
<td>mstr10.tzp</td>
<td>MicroStrategy Mobile Server</td>
</tr>
<tr>
<td>mstr11.tzp</td>
<td>MicroStrategy Portlets, which is a component of MicroStrategy Web</td>
</tr>
<tr>
<td>mstr12.tzp</td>
<td>MicroStrategy GIS Connectors, which is a component of MicroStrategy Web</td>
</tr>
<tr>
<td>mstr13.tzp</td>
<td>All MicroStrategy components and products</td>
</tr>
<tr>
<td>mstr14.tzp</td>
<td>MicroStrategy Enterprise Manager</td>
</tr>
<tr>
<td>mstr15.tzp</td>
<td>MicroStrategy MicroStrategy Identity Server</td>
</tr>
<tr>
<td>mstr16.tzp</td>
<td>MicroStrategy MicroStrategy Identity Manager</td>
</tr>
</tbody>
</table>
### Installation File | MicroStrategy Components That Require The Installation File
--- | ---
mstr18.tzp | MicroStrategy MicroStrategy Communicator
mstr20.tzp | MicroStrategy Messaging Services
mstr21.tzp | MicroStrategy Library Web & Mobile
mstr22.tzp | MicroStrategy Collaboration Server
mstr25.tzp | Node.js which is installed with Intelligence Server and Collaboration Server
mstr26.tzp | Application Schema Server (Early Adopter)
mstr27.tzp | Redis which is installed with MicroStrategy Platform Analytics
mstr28.tzp | MicroStrategy Platform Analytics
mstr29.tzp | Common Files
mstr30.tzp | Community Connectors
mstr31.tzp | Certificate Manager

- Provide the location of the installation files using an `options.txt` file. This lets you access the installation files stored on a folder or stored at a URL and accessed using HTTP or HTTPS. Details on using an `options.txt` file as part of an installation are provided in *Chapter 10, Automated Installation on Linux* and the parameters used to specify the location of the required installation files are described in *Install on Demand Options, page 677*.

If the files required for the installation are stored in a folder, you can instead click **Browse** to navigate to and select the folder that stores the installation files. If all the required installation files are provided in the folder you select, you can click **Enter** to continue the MicroStrategy installation.
Linux Install Wizard - Missing Requirements

This dialog appears if there are system requirements that are not met to install the MicroStrategy products you selected. Review the list of requirements to determine if you can proceed with the installation, or if the installation must be cancelled.

If you are installing MicroStrategy Intelligence Server on Linux, you may see a warning about the value for the Linux kernel setting `vm.max_map_count`. For information about this setting and the recommendation for its value, see *Supporting Intelligence Server memory allocation on Linux, page 81*.

To improve the performance of MicroStrategy Intelligence Server for large scale production applications, Intelligence Server can be configured to use shared memory resources. If a semaphore configuration warning appears, some system resource limits are not configured to fully support the use of shared memory resources.

To support this configuration, cancel the installation and see the limit recommendations in *Configuring shared memory resources, page 79*.

Linux Install Wizard - System Requirements

This dialog appears if the machine you are installing Intelligence Server on does not use the recommended system resource limits to support the use of shared memory resources. It is recommended that you exit the installation and configure these system settings to support shared memory resources. For information on this requirement and the options available to complete the installation, see *Configuring shared memory resources, page 79*. 
Linux Install Wizard - MicroStrategy Identity Server Settings:
Step 1

The MicroStrategy Identity Server installs a database which is a system of record for individual user identities. Use this page to provide the configuration parameters for the MicroStrategy Identity Server to communicate with the database.

Specify the location of the **Tomcat Directory**. The Installation Wizard validates that the directory exists and contains the correct version of Tomcat, and also that you can write to the `webapps` subfolder. Installing the MicroStrategy Identity Server includes a `ROOT.war` file. You can use this `.war` file to deploy your MicroStrategy Identity Server on your Tomcat web application server.

Define the database connection information for the database that stores the MicroStrategy Identity Server database, using the following settings:

- **Server**: The IP address for the machine that hosts the database.
- **Port**: The port number for the database connection. The default port is 3306.
- **User Name**: The account name for the database user that administers the database.
- **Password**: The password for the database user specified above.
- **Server Database Instance**: The name of the MicroStrategy Identity Server database.
- **Log Database Instance**: The name of the database that stores log information for the MicroStrategy Identity Server.
- **Enable SSL Certificate**: Selecting this checkbox allows you to communicate via SSL by default.
To test the database connection, click the **Test** button. The Installation Wizard validates that:

- A connection can be made using the provided server, port, user and password information.
- If the server and log instances exist, the instances are either empty or can be dropped.
- If the server and log instances do not exist, that the provided user has the correct privileges to create the instances.
- The provided user has the correct privileges to create tables.

**Linux Install Wizard - MicroStrategy Identity Server Settings: Step 2 for Ports and Certificates**

The MicroStrategy Identity Server Settings dialog allows you to set up a trust relationship for the MicroStrategy Identity Server using the Public Key Infrastructure (PKI).

Define your MicroStrategy Identity Server's HTTPS port using the following settings:

- **Server (one-way SSL) authentication only**: The default port is **1443**.

- **Client and Server (two-way SSL) mutual authentication**: The default port is **2443**.

Provide the location of your MicroStrategy Identity Server's certificate files using the following settings:

- **SSL Certificate File**: The MicroStrategy Identity Server SSL/HTTPS certificate is used to encrypt the data and enforce authentication. The file contains the certification part of the signed certificate, as well as its public key, in a *crt* format. The default path is *USHER_*
Private Key File: The MicroStrategy Identity Server SSL/HTTPS Private Key file is used to decrypt the data and enforce authentication. The file contains the key part of the signed certificate in a key format. The default path is `USHER_SERVER_INSTALL_PATH/usherApps/shardIDM/conf/Server_ca.key`.

SSL Certificate Chain: The MicroStrategy Identity Server Certificate Authority Chain includes all the certificate authority (CA) certificates that the MicroStrategy Identity Server will trust. The file contains the information in PEM format. The default path is `USHER_SERVER_INSTALL_PATH/usherApps/shardIDM/conf/Server_ca.pem`.

The `USHER_SERVER_INSTALL_PATH` is `INSTALL_PATH/Usher/UsherServer` where `INSTALL_PATH` is the path specified for the install directory.

Linux Install Wizard - MicroStrategy Identity Server Settings: Step 3 for Gateways

You see the MicroStrategy Identity Server Settings dialog if you are installing MicroStrategy Identity Server. This page allows you to set up a trust relationship for the Agent Gateway using the Public Key Infrastructure (PKI).

The Agent Gateway is a component on MicroStrategy Identity Server that is used to synchronize your users from Microsoft Active Directory. The Agent Gateway:

- Establishes a trusted relationship between MicroStrategy Identity Server and the MicroStrategy Identity Agent for Microsoft Active
Directory, which is the application that communicates between your Active Directory server and the MicroStrategy Identity Server.


Provide the **Agent Gateway (one-way SSL) Authentication Only Port** (the default is 9501).

**Linux Install Wizard - MicroStrategy Identity Manager Settings**

Specify the location of the **Apache Directory**. The Install Wizard validates that the directory exists and contains the `conf` and `conf.d` folders, and also that you can write to the `conf.d` subfolder.

Specify a valid **Apache User** to use to access the Apache directory.

MicroStrategy Identity Manager installs a database to manage user identities. You can choose to use the same database connection as MicroStrategy Identity Server, or define the database connection using the following settings:

- **Server**: The IP address for the machine that hosts the database.
- **Port**: The port number to use to connect to the MicroStrategy Identity Manager machine. The default port is 3306.
- **User Name**: The account name for the database user that administers the database.
- **Password**: The password for the database user specified above.
- **Database Instance**: The name of the database instance.
Enable SSL Certificate: Selecting this checkbox allows you to communicate via SSL by default.

To test the database connection, click the Test button.

MicroStrategy Identity Manager is configured as an HTTP connection by default. It is strongly recommended that you configure your connection as an HTTPS connection for security reasons. See your third-party Apache documentation for steps.

Linux Install Wizard - Open Source Software Agreement

If you chose the R component under Intelligence Server, the Open Source Software Agreement appears.

Linux Install Wizard - MicroStrategy Library Web and Mobile

Provide the installation directory for Apache Tomcat to deploy MicroStrategy Library Web and Mobile.

Linux Install Wizard - MicroStrategy Platform Analytics

If Platform Analytics is chosen for installation you need to identify the MySQL database to store the Platform Analytics telemetry data.

Provide the following information in the designated fields:

- **Server**: IP Address
- **Port**: Port number used by the MySQL database
- **Username**
- **Password**

The database user must have all privileges to the database where the Platform Analytics warehouse will be created.
For post-installation configuration instructions, see the Platform Analytics help.

Linux Install Wizard - MicroStrategy Messaging Services Cluster Configuration

If Intelligence Server or Platform Analytics are chosen for installation, the option to configure a Messaging Services cluster is presented.

If no cluster needs to be configured, click Next.

To configure a cluster:

1. Select the I need to create a Messaging Services cluster for Platform Analytics check box.

2. Local Node: Provide the address for the local node.

3. Remote Nodes: Provide a comma separated list of hostnames for remote nodes in the cluster. The node ID is determined by alphabetical order in node list.

See, MicroStrategy Messaging Services for detailed configuration information.

Linux Install Wizard - Topology Configuration

Topology allows administrators to monitor MicroStrategy Services and manage them in MicroStrategy Workstation. The following functionality is supported:

- **Service Health Monitor**
  The administrator can monitor MicroStrategy services through Workstation and visualize if they are running or stopped.

- **Service Start/Stop**
The administrator can start or stop MicroStrategy services through Workstation. This functionality is currently limited to SSH authentication with username and password.

See the topic, *How to View Environment Topology and Monitor Services* in the Workstation Online Help for more information about using topology within Workstation.

The two following types of lightweight agents are used to monitor services in Workstation:

- **Monitoring Agents**

  The monitoring agents come with the installation of most MicroStrategy services. Use them to monitor services and view their health status.

- **Communication Agents:**

  Some monitoring agents also act as communication agents. In addition to monitoring services, communication agents help other monitoring agents locate each other and gather monitoring information. The administrator must pick at least one agent to act as a communication agent, so all other agents can locate each other through this agent.

  On environments where three or more machines host MicroStrategy Services, it is recommended that you choose at least three communication agents to provide redundancy and improved reliability, in case one communication agent becomes unavailable. The machines chosen to act as communication agents must be machines that host MicroStrategy Services, such as Intelligence Server or MicroStrategy Library. If you decide to use only one communication agent, it is recommended that you choose the machine that houses MicroStrategy Library (if deployed through the MicroStrategy installation), since Library communicates directly with Workstation.
If your Library deployment is done through a WAR file or if you do not use a machine that houses MicroStrategy Library to host the communication agent, then use a machine that houses Intelligence Server.

MicroStrategy uses Consul technology for Services Registration. A monitoring agent corresponds to a Consul agent in client mode. A communication agent corresponds to a Consul agent in server mode.
1. On the Topology Configuration dialog of the Installation wizard, select whether your environment has **Single** or **Multiple** machines.

2. If your environment contains multiple machines, determine which machines in your environment you want to act as communication agents. Among all the machines in your environment that will host MicroStrategy Services, select at least one of them (three are recommended for multiple machine environments) to act as a communication agent. Enter each machine in the text field, separated by semicolons. You must input the exact same list on
all machines, using the Topology Configuration dialog of the Installation wizard, including the monitoring and communication agent machines.

The MicroStrategy Services are as follows:

- MicroStrategy Intelligence Server
- MicroStrategy Web Universal
- MicroStrategy Library
- MicroStrategy Mobile
- MicroStrategy Messaging Services
- MicroStrategy Platform Analytics
- MicroStrategy Certificate Store
- MicroStrategy Identity

⚠️ An odd number of communication agents is required due to the leadership selection algorithm.

Examples:

servername1.domain.com;servername2.domain.com;
servername3.domain.com;
servername1.domain.com;

Example: Your environment contains the following two machines and you select one machine to host the communication agent:

- Machine 1 hosts MicroStrategy Library and the communication agent
- Machine 2 hosts the Intelligence Server
When performing the installation, enter the full domain name, \texttt{machine1.domain.com}, on both machines.

Example: Your environment contains the following three machines and you want all machines to host a communication agent:

- Machine 1 hosts MicroStrategy Library
- Machine 2 hosts Intelligence Server 1
- Machine 3 hosts Intelligence Server 2

When performing the installation, enter \texttt{machine1.domain.com;machine2.domain.com;machine3.domain.com;} on all three machines.

Example: Your environment contains the following five machines and you select three machines to host the communication agents:

- Machine 1 hosts MicroStrategy Library
- Machine 2 hosts Intelligence Server 1 and a communication agent
- Machine 3 hosts Intelligence Server 2 and a communication agent
- Machine 4 hosts Intelligence Server 3 and a communication agent
- Machine 5 hosts Intelligence Server or other services

Enter \texttt{machine1.domain.com;machine2.domain.com;machine3.domain.com;} on all machines.

Some important information to keep in mind:
In MicroStrategy 2019, OpenSSH is no longer installed on Windows machines. You must install Open SSH to start and stop services using topology. Upgrading your system from 11.0 to 2019 removes Open SSH.

If consul is already installed on a machine, prior to installing MicroStrategy products, the service registration auto configuration and execution is affected. We do not recommend a separate consul installation on the same machine.

- If you select **Multiple machine environment**, you must enter an odd number of machines.

- If your environment includes more than one machine and machines with dynamic IP addresses as communication agents, use FQDN for the communication agent machine list during the installation.

- If your environment includes more than one machine and machines with more than one IP address as communication agents, it is recommended that you use FQDN for the communication agent machine list during the installation. If IP address list is used, make sure the machine with multiple IP addresses can be pinged with an IP address included in the list.

  Topology supports installation on machines with public IP addresses. To avoid unexpected joins to topology nodes, possibly exposing service information, you must configure your firewall correctly. Your firewall should allow ports for Server RPC (default 8300), and Serf LAN (default 8301), and Serf WAN (default 8302) only for the nodes within the Workstation topology node cluster. See *Enabling Topology Communication Through a Firewall* for more information.

After installing the topology feature, see *Installing and Configuring Topology* for additional instructions on configuring this feature to work in your environment.
Linux Install Wizard - Register as a Service

To enable MicroStrategy processes to start automatically after system start up, you can choose to register the processes as a service.

Linux Install Wizard - CPU License Information

This dialog appears if the Intelligence Server license has a CPU number limitation. Specify the number of CPUs that Intelligence Server is licensed to use.

Linux Install Wizard - Software Activation

If you have installed one or more MicroStrategy server products, you can request an activation code for your MicroStrategy server products upon completion of the installation process. The next few dialogs of the installation process assist you in providing the information needed to request an activation code. MicroStrategy server products include:
• MicroStrategy Intelligence Server

• MicroStrategy Web

• MicroStrategy Mobile Server

This dialog includes a welcome statement for the software activation process. Read the welcome statement and click **Next** to proceed.

**Server Information**

Specify information about your MicroStrategy server installation. Enter the following characteristics:

• **Name**: Distinguishes the name of this MicroStrategy server product installation from any other MicroStrategy server product installations in your company.

• **Location**: Physical location of the machine on which MicroStrategy server products are installed.

• **Use**: Description of how the server is used.

• Click **Privacy Statement** to view the MicroStrategy Privacy Statement.

**Installer Information**

Specify contact information of the person installing the software. After your installation is complete an email containing the Activation Code is sent to the email address you confirm in this software activation step. Enter the following installer information:

• Specify whether you are an employee of the licensed company or installing on behalf of the licensed company.

• Enter the necessary data into all text fields. Make sure the email address you enter is correct. This email address is the recipient of the Activation Code.
Select the check box at the bottom of the page to receive notifications about product updates, events, and special offers from MicroStrategy.

Click Privacy Statement to view the MicroStrategy Privacy Statement.

Contact Information
You see this page if you indicated that you are not an employee of the company licensed to use this software, and are installing the software on behalf of that company.

Specify contact information for the employee license to use the software. Enter the necessary data into all text fields. Make sure the email address you enter is correct. After your installation is complete an email containing the Activation Code is sent to the email address you confirm in this software activation step.

Select the check box at the bottom of the page to receive notifications about product updates, events, and special offers from MicroStrategy.

Click Privacy Statement to view the MicroStrategy Privacy Statement.

Request Activation Code
This page includes options to request an Activation Code now or at a later time. This page provides the following options:

- Select Yes, I want to request an Activation Code and click Next to request an Activation Code. The Activation Code is sent to the email addresses supplied in the Installer Information and Contact Information pages.

- Select No, I will request the Activation Code at a later time and click Next to request an Activation Code at a later time.
If you choose to request an Activation Code at a later time, a message is displayed that instructs you how to request an Activation Code after the installation procedure is completed. For more instructions on requesting an Activation Code at a later time, see Request an Activation Code, page 341 in Chapter 5, Activating Your Installation.

You have a grace period of 30 calendar days to activate your installation. If you do not complete the activation before the grace period expires, your MicroStrategy product stops functioning until you activate it. If you wait to activate your installation, you receive periodic reminders.

Once you request an Activation Code, an email is sent to the email addresses you specify in the Installer Information and Contact Information pages of the software activation procedure. The email provides instructions on how to use the requested Activation Code to activate your software. To activate your installation, you can also use the steps given in Activate Your Installation, page 344 in Chapter 5, Activating Your Installation.

Linux Install Wizard - Start Installer Operation

This dialog provides a description of what configurations are to be completed. If you chose to install, repair, or upgrade MicroStrategy components, this includes listing locations in which the products will be installed (target directories), the location of the installation log file, and license details. If you chose to uninstall MicroStrategy components, this includes a listing of the components to be uninstalled.

When you proceed from this step, the installation process begins, which can take several minutes depending on your computer's hardware configuration.
Linux Install Wizard - MicroStrategy Install Wizard Complete

When the MicroStrategy installation has completed, you can select the following:

- Run MicroStrategy Identity Configuration, which is available if the MicroStrategy Identity Manager has been installed.
- View the Readme for the latest updates.
- Run the MicroStrategy Configuration Wizard, which allows you to configure your MicroStrategy production environment. For more information, see Chapter 6, Configuring and Connecting Intelligence Server.

Click Finish to complete the installation.

Unique Post-Installation Configurations

MicroStrategy supports many different Linux environments with various system configurations. There are a few cases in which you must perform some manual configurations to support the use of MicroStrategy on your system.

Create Links for Intelligence Server Startup in SUSE Linux

If you are installing Intelligence Server on a SUSE Linux environment, you must manually create links for some system files. If you do not create these links, Intelligence Server cannot start correctly.

You need root permissions to access the files and create the necessary links described in this section.

To manually create links for Intelligence Server startup

1. In a console window, browse to the system folder /usr/lib64.
2. In a console window, create the link of libssl.so.4 to libssl.so.0.9.7 with the following command:
   `ln libssl.so.0.9.7 libssl.so.4`

3. In a console window, create the link of libcrypto.so.4 to libcrypto.so.0.9.7 with the following command:
   `ln libcrypto.so.0.9.7 libcrypto.so.4`

Supporting fonts for documents, exported reports, and graphs

When Intelligence Server is running on a Linux platform, all fonts are converted to the Courier New font for:

- Reports exported to PDF format
- Report Services documents
- Graphs contained in HTML documents
- Graphs displayed in MicroStrategy Web

This occurs because the fonts required by the PDF component are missing from Linux machines running Intelligence Server.

- MicroStrategy cannot package these fonts with Intelligence Server due to licensing restrictions.

For steps to support fonts such as Microsoft True Type fonts for the MicroStrategy features listed above, see Setup for Executing Report Services Documents, page 553.

Starting and Stopping Intelligence Server Kafka service

For Linux users, the Kafka Consumer will be installed and automatically running after installation or upgrade. You will find KafkaConsumer.sh, LogConsumer.properties, and
KafkaConsumer.jar under <install_path>/IntelligenceServer/KafkaConsumer. You can use the following commands to control the Kafka Consumer logging activity:

- **Start service**: `<path_to>/KafkaConsumer.sh start`
- **Stop service**: `<path_to>/KafkaConsumer.sh stop`
- **Restart service**: `<path_to>/KafkaConsumer.sh restart`
- **Check service status**: `<path_to>/KafkaConsumer.sh status`

**Starting and Stopping the PDF Exporter Service**

For Linux users the PDF Exporter Service will not be started when OS is restarted. To start this service manually, run the `pdfexporter.sh` script found under the path `install/IntelligenceServer/PDFExportService`. Commands for this script include:

- `<path_to>/pdfexporter.sh start` to start the PDF Exporter Service.
- `<path_to>/pdfexporter.sh stop` to stop the PDF Exporter Service.
- `<path_to>/pdfexporter.sh restart` to restart the PDF Exporter Service.
- `<path_to>/pdfexporter.sh status` to check the status the PDF Exporter Service.

For other configuration settings, see *Export Engine Configuration*.

**Verifying Installation**

During installation, the MicroStrategy Installation Wizard gathers and records information about your system and your installation
selections. You can verify the setup information through the installation log file (install.log).

By default, the log file is located in INSTALL_PATH where, INSTALL_PATH is the directory you specified as the install directory in the MicroStrategy Installation Wizard.

The log file includes information about the following:

- Installation date
- Target directory
- Operating system identification
- Selected installation options
- Selected licensing details

This log file can be helpful if you encounter errors during the installation process. The log file records the reasons for errors occurring.

Directory Structure

The following table lists the directories in which MicroStrategy files are installed:

<table>
<thead>
<tr>
<th>PATH/Directory</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOME_PATH</td>
<td>Configuration files that can be modified after installation.</td>
</tr>
<tr>
<td>HOME_PATH/env</td>
<td>Scripts to set up the proper environment for the MicroStrategy applications. If additional products are installed on the same machine at a later time, backups of the original scripts are saved here.</td>
</tr>
<tr>
<td>HOME_PATH/bin</td>
<td>Scripts to launch the MicroStrategy applications.</td>
</tr>
<tr>
<td>INSTALL_PATH</td>
<td>Files that are not supposed to change after the installation.</td>
</tr>
</tbody>
</table>
### PATH/Directory | Contents
--- | ---
**INSTALL_PATH/CommandManager** | installation is complete.
**INSTALL_PATH/Help** | MicroStrategy Command Manager files. This is the default directory for Command Manager but another location can be selected during installation.
**INSTALL_PATH/GISConnectors** | Documentation and Help for MicroStrategy products.
**INSTALL_PATH/IntelligenceServer/bin** | MicroStrategy Portlet files. This is the default directory for the Portlets but another location can be selected during installation.
**INSTALL_PATH/Mobile** | Intelligence Server-specific binary files.
**INSTALL_PATH/PDFGeneratorFiles** | MicroStrategy Mobile and Mobile Server JSP files. This is the default directory for Mobile but another location can be selected during installation.
**INSTALL_PATH/ReleaseNotes** | Support files (fonts) for the PDF generation feature of Intelligence Server.
**INSTALL_PATH/Portlets** | MicroStrategy Readme for this release of MicroStrategy products.
**INSTALL_PATH/SystemManager** | MicroStrategy Portlet files. This is the default directory for the Portlets but another location can be selected during installation.
**INSTALL_PATH/Usher** | MicroStrategy System Manager files. This is the default directory for System Manager but another location can be selected during installation.
**INSTALL_PATH/WebServicesJ2EE** | MicroStrategy Identity files.
**INSTALL_PATH/WebUniversal** | MicroStrategy Web Services deployment path.
**INSTALL_PATH/_jvm** | MicroStrategy Web deployment path.
**INSTALL_PATH/_jvm** | The Java Runtime Environment (JRE) to be used by
Configuring your MicroStrategy Installation

After completing the steps to install MicroStrategy products, you can set up and configure your installation. To help guide the rest of your installation and configuration steps, refer to the section *Installing and Configuring MicroStrategy on Linux, page 112* in *Chapter 1, Planning Your Installation*, for an installation and configuration checklist.

To configure MicroStrategy Identity components, refer to the *MicroStrategy Identity Help* guide.
Installing and Configuring MicroStrategy Library

The MicroStrategy Library installation includes the MicroStrategy REST Server as well as the MicroStrategy Collaboration Server, which enable the following functionalities:

- **Browse shared content**: Users have access to their own library where shared dossiers and documents are available to be consumed.

- **Search for dossiers**: Users can easily search for dossiers with matching criteria such as dossier name, chapter/page name, etc.

- **Manipulate the data view with visualizations and filters**: Users can consume the data in a dossier with visualization and filter manipulations. Every action is recorded in the metadata so that a user can resume from the last stopping point for the next login.

- **Share dossier to other users**: Users can share their dossier view to other users. All of the manipulations will be carried over so that they consume the data based on same context.

- **Interact with other users via comments and notifications (Collaboration Server)**: User can leave commentary, tag other users, and embed filters to collaborate with other users within any dossier.

The Collaboration Server also supports the following advanced functionalities:

- **Realtime monitoring of connected users**: Every user that connects to the Library Web or Mobile application is tracked by this service.
● **Realtime monitoring of user comments**: Users can create comments associated with each individual page. Users will receive new comments as they arrive without the need for refreshing. Persistence of comments. Comments persist for retrieval at a later time.

● **Comments can contain "embedded objects"**: Users can store filter settings with a comment for another user to apply.

● **User mention and notification**: Users can send notification to another user via user mention. In addition to "in-app" notification, users can also be notified by APNS (device notification) and email notification.

### MicroStrategy Library Architecture

The following terminology will help you to understand what is behind the Collaboration Server.

● **Web Socket** is a protocol providing full-duplex communication channels over a single TCP connection. Web Socket is supported with all major browsers and can be used in mobile applications. For Web Socket to work in an application, an application server (web server) needs to support it as well.

   If the Library cannot establish a Web Socket connection due to a blocked protocol in an environment that contains load balancers, it attempts an HTTP connection and uses a technique called "long polling" to simulate a real time message exchange.

● **Node.js** is an open-source, cross-platform runtime environment for developing server-side Web applications. Node.js is the runtime environment for hosting the MicroStrategy Collaboration Server.

● **MongoDB** is an open-source document-oriented database used to store collaboration data for the MicroStrategy Library application.
The following diagram shows the Library architecture, including the Collaboration Server, shown in blue.

Messages flow in both directions of a Web Socket connection. The MicroStrategy Collaboration Server manages which messages go to which clients based on a publish-subscribe model. A client application may publish a comment to a topic, and all clients subscribed to the topic will be notified.
MicroStrategy Library System Requirements

Collaboration Server

- Node.js 8.11.1 (Installed with MicroStrategy)
- MongoDB 4.0.0 (Installed with MicroStrategy)

Library Administration Control Panel

You can access the Library Control Panel by navigating to 
<FQDN>:<port>/MicroStrategyLibrary/admin. The control panel allows you to examine and configure settings for Library Server and Collaboration Server. Additionally you can view the settings related to communication with the Intelligence Server and configurations in MicroStrategy Library such as the Intelligence Server cluster it is connected to.

Overview Panel

The Library Admin Overview panel provides the ability to examine the machine name, port, state, and Intelligence and Collaboration Servers. The top right corner of each component displays its current status. Click Edit to change settings for Intelligence Server and Collaboration Server. You can also choose to expose or hide the Collaboration Service feature to users.
Configure Collaboration Server

To add a Collaboration Server, only the machine name (hostname) and port are required.

The Collaboration Server uses 3000 as the default port. If the administrator uses 0 as the port number when configuring the Collaboration Server, it will be replaced with 3000.

Configure Intelligence Server

To add an Intelligence Server, the administrator needs to specify the machine name (hostname), port, and TLS status.

The Intelligence Server uses 34952 as the default port. If the administrator uses 0 as the port number, the Library Server will try to connect to an Intelligence Server under port 34952.

Error / Warning Messages / Troubleshooting

Existing configuration issues will be displayed as either an error icon or a warning icon in the Overview page:
- **Error icons in red**: The system is not running as expected. Click to view a more detailed message.

- **Warning icons in yellow**: There are some minor issues that should be addressed.

When you click an error or warning icon, a pop up window will display a detailed error message and provide some actions/links regarding how the issue happened and how to resolve that particular issue.

---

**Timeout Settings**

If you receive a Timeout Error in the Control Panel, modify the timeout settings in the Library Server configuration file `configOverride.properties` to a larger value. For example:

```
http.connectTimeout=5000
http.requestTimeout=10000
```

---

**Library Server Tab**

The Library Server tab allows an administrator to view their deployment's Library URL, Web Server URL, configure authentication
modes, security settings, and access the Mobile configuration link.

MicroStrategy Web

This provides a connection URL to MicroStrategy Web, which enables the Web options in Library.

Authentication Modes

The Authentication Modes section gives an administrator to dictate which authentication modes they want to allow. When the authentication mode is updated and saved, the server typically requires a restart.

- Trusted authentication mode cannot be used in combination with any other log in mode.

Security Settings

The first security settings section allows administrators to allow Library embedding in other sites. In other words, this section enables Cross-Origin Resource Sharing (CORS) A mechanism that uses additional HTTP headers to tell a browser to let a web application
running at one origin (domain) have permission to access selected resources from a server at a different origin. settings. You must enable your CORS settings to use MicroStrategy products like HyperIntelligence or MicroStrategy for Office or to embed a dossier in a website.

To enable CORS, select All. When this security setting is updated and saved, restart the Library application.

Additionally, the security settings section allows administrator to enter an optional secret key which is used to allow sharing sessions between MicroStrategy Web and MicroStrategy Library.

Mobile Configuration

The mobile configuration link provides with the mobile server link which can copied and can be accessed.

Intelligence Server Tab

The Intelligence Server tab allows the administrator to manage the Intelligence Server Machine information.

It includes the following sections:
• **Intelligence Server Machine Information**

  This section allows for the input of Machine name and Port Number for the Intelligence Server.

• **Intelligence Server Connection Settings**

  The Intelligence Server Connection Settings tab allows the administrator to configure the following parameters:

  • Initial Pool Size
  
  • Maximum Pool Size
  
  • Request Timeout (socket connection request timeout is in milliseconds for the Intelligence Server)
  
  • Working Set Size (indicates the number of recent reports or documents in memory for manipulation)
  
  • Search Working Set Size (Indicates the maximum number of concurrent searches which stay in memory)

• **Collaboration Server Panel**

  This panel displays the server URL and that state of the Collaboration Server machine. It also allows the administrator to update the following settings for the Collaboration Server that is currently connected:

  • Enable logging
  
  • Enable TLS
  
  • Trusted Certificate Setting

  In order to access this section, the communication between the Library Server and the Collaboration Server needs to be established first: no errors or warnings from the Library Server to the Collaboration Server on the overview page.
Collaboration Server Settings

**Enable Logging**

This setting can enable / disable diagnostic logging functionality in the Collaboration Server.

**Enable TLS**

Selecting this setting requires a keystore path and a passphrase.

Any changes to this section require the administrator to restart the Collaboration Server manually. Related warnings will be displayed on this page until the restart is performed.

**Trusted Certificate Setting**

This section is only visible when the Collaboration Server targets a TLS enabled Library Server.

**Installing MicroStrategy Library on Windows**

Perform the following steps to install MicroStrategy Library on a Windows server.

1. Run the MicroStrategy installer by clicking the `setup.exe` application from the extracted zip file.

2. To install the MicroStrategy Library components, select the following features when prompted.
   - MicroStrategy Library
     - MicroStrategy Library Web & Mobile
     - MicroStrategy Collaboration Server
3. Click **Next**

   If the install does not complete successfully, review the **install.log** file for errors. The log file is located in the MicroStrategy Common Files directory `C:\Program Files (x86)\Common Files\MicroStrategy\install.log`.

**Deployment Scenarios**

After the installation of the MicroStrategy Library, applicable default deployment will be performed automatically. You may need to further configure the MicroStrategy Library Web/Mobile Server and MicroStrategy Collaboration Server, depending on your environment and requirements. The following sections describe different deployment scenarios:

Diagrams in the following sections include the following MicroStrategy components:
• **Library Web application**

A single page application (SPA) that runs inside a web browser. It is populated by an initial connection to the MicroStrategy Library Server.

• **Library Mobile application**

A native mobile application that interacts with the same MicroStrategy Library Server.

• **MicroStrategy Library Server**

A Java-based HTTP web service that handles request for data from the MicroStrategy Library Web, MicroStrategy Mobile, and Workstation applications.

• **MicroStrategy Collaboration Server**

A Node.js application that establishes a persistent Web Socket connection with the MicroStrategy Library Web/Mobile application. It interacts with a Collaboration Server repository to persist collaboration data and with the MicroStrategy Library Server as well.

• **MicroStrategy Intelligence Server**

A native OS application that handles all requests the metadata repository and Data Warehouse transactions.

Default Windows Deployment

Default Product Folders

• `C:\Program Files (x86)\MicroStrategy\Library`

• `C:\Program Files (x86)\MicroStrategy\Collaboration Server`
• C:\Program Files (x86)\Common Files\MicroStrategy\Tomcat\apache-tomcat-8.5.30\webapps\MicroStrategyLibrary

Default Registered Windows Services

• **Library Server**: Apache Tomcat 8.5 Tomcat8
• **Collaboration Server repository**: MongoDB
• **Collaboration Server**: MicroStrategy Collaboration Server

MicroStrategy Library Server Configuration File

The MicroStrategy Library Server reads configuration settings from the file configOverride.properties. It also provides default values for the configuration in the file configDefaults.properties. The file configDefaults.properties contains a key-value pair of configuration options. To specify a configuration option, the desired key can be copied from the configDefaults.properties and saved into configOverride.properties with a value. For default deployment, the two files are located in:

• C:\Program Files (x86)\Common Files\MicroStrategy\Tomcat\apache-tomcat-8.5.30\webapps\MicroStrategyLibrary\WEB-INF\classes\config\configDefault.properties

• C:\Program Files (x86)\Common Files\MicroStrategy\Tomcat\apache-tomcat-8.5.30\webapps\MicroStrategyLibrary\WEB-INF\classes\config\configOverride.properties

If MicroStrategy Library is installed separately from the Intelligence Server, you need to manually configure the connection information by modifying the file configOverride.properties. These are also
the minimum values which should be specified in configOverride.properties:

- iserver.default.hostname= <host or IP address of Intelligence Server>
- iserver.default.port= <Intelligence Server Port>

If the MicroStrategy Collaboration Server is configured, and the file configOverride.properties is being manually set, the following configuration properties need to be specified:

- services.collaboration.enabled=true
- services.collaboration.baseUrl= <{PROTOCOL://HOSTNAME:PORT} The URL path to the Collaboration Server>
- identityToken.secretKey= <Secret Key phrase (spaces allowed) for Signing web token>

The configDefaults.properties should never be altered because it may cause issues when migrating or upgrading an environment.

If the MicroStrategy Collaboration Server is installed separately from the MicroStrategy Library, you need to manually configure the connection information for Collaboration Server as well, by modifying configuration property file located in: C:\Program Files (x86)\MicroStrategy\Collaboration Server\MicroStrategy\Collaboration Server\config.json.

The property needs to be specified: authorizationServerUrl: http://<FQDN>:<PORT>/MicroStrategyLibrary/api

⚠️ All parameters in the config.json file are case sensitive.
The communication port must be available for Collaboration Server to be started.

The following diagram shows the default deployment:

![Diagram of default deployment]

The communication between components is HTTP, not HTTPS. This deployment is useful for initial testing in a lab environment, but not recommended for production. For a production environment, where security is a priority, it is recommended to implement a secure deployment.

Windows Secure Deployment

A secure deployment is strongly recommended. In a secure deployment scenario, you add HTTPS/TLS support to the Library
application. The server processes will remain on the same machine, but will communicate with each other through a secure channel.

The following diagram shows the secure deployment:

Perform the following procedure to enable secure communication between the Library applications and services on a Windows server. These instructions assume that a default deployment has already been completed, and Intelligence Server has already been TLS enabled.

Prepare Files for Secure Deployment

**Contact your system administrator** to obtain the following files and information:
- A "key store" file in PKCS12 or PFX format which contains the private key and the public certificate for this machine. In the example this file is saved as `keystore.pfx`.

Use this same file for both Collaboration Server and Tomcat.

The Collaboration Server only supports PKCS12 or PFX format.

- The passphrase used to create the key store file.

  The passphrase is needed to read the contents of the key store. In the example, the passphrase is "keystore_password".

- If the server certificate is not signed by a public certificate authority, then collect the Root CA certificate in the chain. The 'certificate' file allows Collaboration Server to open a secure connection to the Library Server.

- A “trust store” file (that can be created from the certificates) and the passphrase used.

  The trust store file is used by the Library Server to open a secure connection to the Collaboration Server.

  This is only required if the certificate is not signed by a public certificate authority.

  In the example, the file is saved as `truststore.pfx` and the passphrase is "truststore_password".

Copy the files obtained in step 1 into the following directory:

C:\Program Files (x86)\MicroStrategy\Collaboration Server

Set Up Secure JEE App Server

Configure Tomcat to use HTTPS to configure Tomcat to be a secure endpoint:
1. Open the following directory:

   C:\Program Files (x86)\Common Files\MicroStrategy\Tomcat\apache-tomcat-9.0.12\conf

2. Open the server.xml file with a text editor.

3. Look for the Connector element whose port is 8443. Ensure it is not commented out. If Connector doesn't exist, add it.

4. Edit the 'keystoreFile' and 'keystorePass' attributes of this element.

   <Connector
   scheme="https" secure="true" SSLEnabled="true"
   keystoreFile="C:\Program Files (x86)\MicroStrategy\Collaboration Server\keystore.pfx"
   keystorePass="<keystore_passphrase>"
   clientAuth="false"
   sslProtocol="TLS"/>

5. Save and close the file.

   You can refer to more detailed instruction from Tomcat documentation.

Set Up Secure MicroStrategy Library Server

Modify the MicroStrategy Library Server settings to open a secure connection to the Collaboration Server:
1. Navigate to the following directory:

   C:\Program Files (x86)\Common\Files\MicroStrategy\Tomcat\apache-tomcat-9.0.12\webapps\MicroStrategyLibrary\WEB-INF\classes\config

2. Edit `configOverride.properties` file:

   ```
   services.collaboration.baseURL=https://<FQDN>:3000
   // set tlsEnabled to true only when the certificate is not assigned by a public certificate authority
   services.collaboration.tlsEnabled=true
   trustStore.path=\WEB-INF\truststore.pfx
   trustStore.passphrase=<truststore_password>
   ```

   All parameters are case sensitive. Read the instructions MicroStrategy Library Server Configuration Properties for more information.

3. Save and close the file.

4. Restart Tomcat Server.

5. Verify that the MicroStrategy Library Server is running:
   - Open a web browser and enter https://FQDN:8443/MicroStrategyLibrary
   - If you see login page, then MicroStrategy Library Server is working.

Set Up Secure Collaboration Server

Modify the Collaboration Server settings:
1. **Edit the config.json file located in:**

   `C:\Program Files (x86)\MicroStrategy\Collaboration Server`

2. Insert the following lines into the file. These reference the security artifacts we created earlier.

   ```json
   "enableTls": true,
   "keystoreFile": "C:\Program Files (x86)\MicroStrategy\Collaboration Server\keystore.pfx",
   "passphrase": "<keystore_passphrase>",
   "trustedCerts": [ "C:\Program Files (x86)\MicroStrategy\Collaboration Server\certificate.pem"
   ```

   All parameters are case sensitive. Read the instructions Collaboration Server Configuration Properties for more information.

3. **Modify the authorizationServerUrl field in the config.json file to specify “https”, the HTTPS port (8443) and replace “localhost” with the fully qualified domain name (FQDN) for this machine.**

   It should look like the following:

   ```json
   authorizationServerUrl:
   https://<FQDN>:8443/MicroStrategyLibrary/api
   ```

4. Make sure the file is still valid JSON. Copy the text into an Internet JSON validator to verify.

5. Save the file.

6. Restart the Collaboration Server.

7. Verify that the Collaboration Server is running:
Open a web browser and enter
https://FQDN:3000/status

If you see a web page (with no errors), then Collaboration Server is working.

Note the following:

- JEE App Server – key store

  This is used to establish the identity of the JEE App Server, which houses the MicroStrategy Library Server. This is changed in a system configuration file. For example, the server.xml file in a Tomcat installation.

- MicroStrategy Library Server – trust store

  Used to verify identity of the Collaboration Server and/or Intelligence Server for outgoing HTTPS/TLS calls when the Server certificate is not signed by a public certificate authority. The trust store is configured in the configOverride.properties file.

- MicroStrategy Collaboration Server – key store / certificate

  The key store is used for trusted communication between the Library Web/Mobile Client and the Collaboration Server. The certificate is used to verify the identity of the Library Server. The key store and certificates are configured in the config.json file.

Manually Deploy Library on a Windows Server

Collaboration Server Setup

If MicroStrategy Library Web is going to utilize collaborating features, perform the following steps to manually configure the Collaboration Server.
1. Browse to the Collaboration Server installation directory and open the `config.json` file.

   C:\Program Files (x86)\MicroStrategy\Collaboration Server

2. Edit the `config.json` file in the following format:

   ```json
   {
     "port": <COLLAB_SERVER_PORT>,
     "dburl": "mongodb://{MONGODB_USERNAME}:{MONGODB_PASSWORD}@{MONGODB_HOSTNAME}:${MONGODB_PORT}/MONGODB_DB_NAME}?authSource=admin",
     "authorizationServerUrl": "{PROTOCOL}://{LIBRARY_HOSTNAME}:${LIBRARY_PORT}/MicroStrategyLibrary/api",
     "secretKey": "SOME SECRET KEY"
   }
   ```

   You may need to configure more settings in the `config.json` file. See Configuring Collaboration Server for more information.

3. Restart Collaboration Server, and access `{PROTOCOL}://{COLLAB_SERVER_HOSTNAME}:${COLLAB_SERVER_PORT}/status` to verify.

**Deploy Library Web App**

The following steps assume Tomcat service has been started.

1. Open `<TOMCAT_INSTALLATION_DIRECTORY>\webapps`, and paste the copied MicroStrategyLibrary.war file.

2. Once the MicroStrategyLibrary folder is created, access Library Administration Control Panel:

   `{PROTOCOL}:/<FQDN>:<port>/MicroStrategyLibrary/admin` with Tomcat administrator credentials.
3. In the Library Server Panel, under Security Settings, create a 'secret key'.


5. Restart Tomcat service, and access
   {PROTOCOL}://<FQDN>:<port>/MicroStrategyLibrary to verify.

Configure Logging for Windows Deployment

For default deployment, log files are located in the following default locations:

- **Install log**: C:\Program Files (x86)\Common Files\MicroStrategy\install.log
- **Library Server log**: C:\Program Files (x86)\Common Files\MicroStrategy\Log\MicroStrategyLibrary\MicroStrategyLibrary.log
- **Collaboration Server log**: C:\Program Files (x86)\Common Files\MicroStrategy\Log\collabsrvr_nodejs.log
- **MongoDB log**: C:\Program Files (x86)\MicroStrategy\Collaboration Server\MongoDB\data\log\mongod.log

Collaboration Server Advanced Logging

To enable Collaboration Server advanced logging for troubleshooting purposes:

1. Set the logging field to true in the config.json file in C:\Program Files (x86)\MicroStrategy\Collaboration Server
2. Restart the service using the Services application in the Control Panel.

Be sure to turn off the advanced logging after troubleshooting.

Installing MicroStrategy Library on Linux

1. Run the MicroStrategy installer:

   Choose the **setup.sh** application from the extracted zip file.

2. Follow the instruction in the installation wizard:

   To install the MicroStrategy Library components, select the following features when prompted:

   - MicroStrategy Library
     - MicroStrategy Library Web & Mobile
     - MicroStrategy Collaboration Server
3. Click **Next**.

4. After selecting components to install, specify the Apache Tomcat Directory, for example:

![MicroStrategy Installer for Unix/Linux](image)

If the install does not complete successfully, review the `install.log` file for errors. The log file is located in `<HOME_PATH>/log/install.log`.

**Deployment Scenarios**

After the installation of the MicroStrategy Library, applicable default deployment will be performed automatically. You may need to further configure the MicroStrategy Library Web/Mobile Server and MicroStrategy Collaboration Server, depending on your environment and requirements. The following sections describe different deployment scenarios:

Diagrams in the following sections include the following MicroStrategy components:
• **Library Web application**
  
  A single page application (SPA) that runs inside a web browser. It is populated by an initial connection to the MicroStrategy Library Server.

• **Library Mobile application**
  
  A native mobile application that interacts with the same MicroStrategy Library Server.

• **MicroStrategy Library Server**
  
  A Java-based HTTP web service that handles request for data from the MicroStrategy Library Web, MicroStrategy Mobile, and Workstation applications.

• **MicroStrategy Collaboration Server**
  
  A Node.js application that establishes a persistent Web Socket connection with the MicroStrategy Library Web/Mobile application. It interacts with a Collaboration Server repository to persist collaboration data and with the MicroStrategy Library Server as well.

• **MicroStrategy Intelligence Server**
  
  A native OS application that handles all requests the metadata repository and Data Warehouse transactions.

**Default Linux Deployment**

**Default Product Folders**

• `<INSTALL_PATH>/LibraryWebMobile`

• `<INSTALL_PATH>/CollaborationServer`
• `<TOMCAT_INSTALL_DIRECTORY>/webapps/MicroStrategyLibrary`

Default URLs

• **MicroStrategy Library Web application:**
  http://<FQDN>:8080/MicroStrategyLibrary

• **MicroStrategy Collaboration Server:** http://<FQDN>:3000
  • View the status of **Collaboration Server**:
    http://<FQDN>:3000/status

MicroStrategy Library Server Configuration Files

The MicroStrategy Library Server reads configuration settings from the file `configOverride.properties`. It also provides default values for the configuration in the file `configDefaults.properties`. The file `configDefaults.properties` contains a key-value pair of configuration options. To specify a configuration option, the desired key can be copied from the `configDefaults.properties` and saved into `configOverride.properties` with a value. For default deployment, the two files are located in:

• `<TOMCAT_INSTALL_DIRECTORY>/webapps/MicroStrategyLibrary/WEB-INF/classes/config/configDefault.properties`

• `<TOMCAT_INSTALL_DIRECTORY>/webapps/MicroStrategyLibrary/WEB-INF/classes/config/configOverride.properties`

If MicroStrategy Library is installed separately from the Intelligence Server, you need to manually configure the connection information by modifying the file `configOverride.properties`. These are also
the minimum values which should be specified in configOverride.properties:

- iserver.default.hostname = <host or IP address of Intelligence Server>
- iserver.default.port = <Intelligence Server Port>

If the MicroStrategy Collaboration Server is configured, and the configOverride.properties is being manually set, the following configuration properties need to be specified:

- services.collaboration.enabled = true
- services.collaboration.baseURL = <{PROTOCOL://HOSTNAME:PORT} The URL path to the Collaboration Server>
- identityToken.secretKey = <Secret Key phrase (spaces allowed) for Signing web token>

The configDefaults.properties should never be altered because it may cause issues when migrating or upgrading an environment.

If the MicroStrategy Collaboration Server is installed separately from the MicroStrategy Library, you need to manually configure the connection information by modifying configuration property file located in <INSTALL_PATH>/CollaborationServer/config.json.


All parameters in the config.json file are case sensitive.

The communication port must be available for Collaboration Server to be started.
The communication between components is HTTP, not HTTPS. This deployment is useful for initial testing in a lab environment, but not recommended for production. For a production environment, where security is a priority, it is recommended to implement a secure deployment.

**Linux Secure Deployment**

A secure deployment is strongly recommended. In a secure deployment scenario, you add HTTPS/TLS support to the Library application. The server processes will remain on the same machine, but will communicate with each other through a secure channel.
Perform the following procedure to enable secure communication between the Library applications and services on a Windows server. These instructions assume that a default deployment has already been completed, and Intelligence Server has already been TLS enabled.

Prepare Files for Secure Deployment

**Contact your system administrator** to obtain the following files and information:

- A “key store” file in PKCS12 or PFX format which contains the private key and the public certificate for this machine. In the example this file is saved as *keystore.pfx*

Use this same file for both Collaboration Server and Tomcat.
The Collaboration Server only supports PKCS12 or PFX format.

- The passphrase used to create the key store file.

  The passphrase is needed to read the contents of the key store. In the example, the passphrase is "keystore_password".

- If the server/leaf certificate is not ultimately signed by a public certificate authority then collect all of the certificates in PEM format. This should include all certificates in the chain. The 'certificate' file allows Collaboration Server to open a secure connection to the Library Server.

- A “trust store” file (that can be created from the certificates) and the passphrase used.

  The trust store file is used by the Library Server to open a secure connection to the Collaboration Server.

  This is only required if the certificate is not signed by a public certificate authority.

  In the example, the file is saved as truststore.pfx and the passphrase is "truststore_password".

Copy the files obtained in step 1 into the following directory:

<INSTALL_PATH>/CollaborationServer

Set Up Secure JEE App Server

Configure Tomcat to use HTTPS to configure Tomcat to be a secure endpoint:

1. Open the following directory:

   <TOMCAT_INSTALL_DIRECTORY>/conf

2. Open the server.xml file with a text editor.
3. Look for the Connector element whose port is 8443. Ensure it is not commented out. If Connector doesn't exist, add it.

4. Edit the 'keystoreFile' and 'keystorePass' attributes of this element.

```xml
<Connector
    port="8443" maxThreads="200"
    scheme="https" secure="true" SSLEnabled="true"
    keystoreFile="<INSTALL_PATH>/CollaborationServer/keystore.pfx"
    keystorePass="<keystore_passphrase>"
    clientAuth="false" sslProtocol="TLS"/>
```

5. Save and close the file.

You can refer to more detailed instruction from Tomcat documentation.

Set Up Secure MicroStrategy Library Server

Modify the MicroStrategy Library Server settings to open a secure connection to the Collaboration Server:

1. Navigate to the following directory:

```bash
<TOMCAT_INSTALL_DIRECTORY>/webapps/MicroStrategyLibrary/WEB-INF/classes/config
```

2. Edit configOverride.properties file:

```properties
services.collaboration.baseURL=https://<FQDN>:3000
// set tlsEnabled to true only when the certificate is not assigned by a public certificate authority
services.collaboration.tlsEnabled=true
```
In allation and Configuration Guide

trustStore.path=\WEB-INF\truststore.pfx
trustStore.passphrase=<truststore_password>

All parameters are case sensitive. Read the instructions MicroStrategy Library Server Configuration Properties for more information.

3. Save and close the file.
4. Restart Tomcat Server.
5. Verify that the MicroStrategy Library Server is running:
   - Open a web browser and enter
     https://FQDN:8443/MicroStrategyLibrary
   - If you see login page, then MicroStrategy Library Server is working.

Set Up Secure Collaboration Server

Modify the Collaboration Server settings:

1. Edit the config.json file located in:
   
   <INSTALL_PATH>/CollaborationServer

2. Insert the following lines into the file. These reference the security artifacts we created earlier.

   "enableTls": true,
   "keystoreFile": "<INSTALL_PATH>/CollaborationServer/keystore.pfx",
   "passphrase": "<keystore_passphrase>",
   "trustedCerts": [ "<INSTALL_PATH>/CollaborationServer/certificate.pem"]
All parameters are case sensitive. Read the instructions Collaboration Server Configuration Properties for more information.

3. Modify the authorizationServerUrl field in the config.json file to specify “https”, the HTTPS port (8443) and replace “localhost” with the fully qualified domain name (FQDN) for this machine.

It should look like the following:

```
authorizationServerUrl:
https://<FQDN>:8443/MicroStrategyLibrary/api
```

4. Make sure the file is still valid JSON. Copy the text into an Internet JSON validator to verify.

5. Save the file.

6. Restart the Collaboration Server.

7. Verify that the Collaboration Server is running:

   - Open a web browser and enter https://FQDN:3000/status

   - If you see a web page (with no errors), then Collaboration Server is working.

Note the following:

- JEE App Server – key store

  This is used to establish the identity of the JEE App Server, which houses the MicroStrategy Library Server. This is changed in a system configuration file. For example, the server.xml file in a Tomcat installation.

- MicroStrategy Library Server – trust store
Used to verify identity of the Collaboration Server and/or Intelligence Server for outgoing HTTPS/TLS calls when the Server certificate is not signed by a public certificate authority. The trust store is configured in the `configOverride.properties` file.

- **MicroStrategy Collaboration Server – key store / certificate**

  The key store is used for trusted communication between the Library Web/Mobile Client and the Collaboration Server. The certificate is used to verify the identity of the Library Server. The key store and certificates are configured in the `config.json` file.

**Manually Deploy Library on a Linux Server**

**Collaboration Server Setup**

If MicroStrategy Library Web is going to utilize collaborating features, perform the following steps to manually configure the Collaboration Server.

1. Browse to the Collaboration Server installation directory and open the `config.json` file.

   `{INSTALL_PATH}/CollaborationServer`

2. Edit the `config.json` file in the following format:

```json
{
    "port": <COLLAB_SERVER_PORT>,
    "dburl": "mongodb://{MONGODB_USERNAME}:{MONGODB_PASSWORD}@{MONGODB_HOSTNAME}:${MONGODB_PORT}/{MONGODB_DB_NAME}?authSource=admin",
    "authorizationServerUrl": "{PROTOCOL}://{LIBRARY_HOSTNAME}:{LIBRARY_PORT}/MicroStrategyLibrary/api",
    "secretKey": "SOME SECRET KEY"
}
```
You may need to configure more settings in the `config.json` file. See Configuring Collaboration Server for more information.

3. Restart Collaboration Server, and access `{PROTOCOL}://{COLLAB_SERVER_HOSTNAME}:{COLLAB_SERVER_PORT}/status` to verify.

Deploy Library Web App

The following steps assume Tomcat service has been started.

1. Open `<TOMCAT_INSTALLATION_DIRECTORY>/webapps`, and paste the copied `MicroStrategyLibrary.war` file.

2. Once the `MicroStrategyLibrary` folder is created, access Library Administration Control Panel:

   `{PROTOCOL}://<FQDN>:<port>/MicroStrategyLibrary/admin` with Tomcat administrator credentials.

3. In the Library Server Panel, under Security Settings, create a 'secret key'.


5. Restart Tomcat service, and access `{PROTOCOL}://<FQDN>:<port>/MicroStrategyLibrary` to verify.

Configure Logging for Linux Deployment

Log files are located in the following default locations:

- Install log: `<HOME_PATH>/log/install.log`

Collaboration Server log: <HOME_PATH>/log/collabsvr_nodejs.log

MongoDB log: <INSTALL_PATH>/CollaborationServer/MongoDB/mongod.log

Collaboration Server Advanced Logging

To enable Collaboration Server advanced logging for troubleshooting purposes:

1. Set the logging field to true in the config.json file in <INSTALL_PATH>/CollaborationServer/config.json.

2. Restart the service using collaborationServer.sh restart.

Be sure to turn off the advanced logging after troubleshooting.

MicroStrategy Library Server Configuration Properties

The configOverride.properties is located in:

Windows: C:\Program Files (x86)\Common Files\MicroStrategy\Tomcat\apache-tomcat-8.5.30\webapps\MicroStrategyLibrary\WEB-INF\classes\config\configOverride.properties

Linux: <TOMCAT_INSTALL_DIRECTORY>/webapps/MicroStrategyLibrary/WEB-INF/classes/config/configOverride.properties
This file includes the following properties related to authentication modes:

<table>
<thead>
<tr>
<th>Property</th>
<th>Specifies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authentication Mode Settings</strong></td>
<td></td>
</tr>
<tr>
<td>auth.modes.available</td>
<td>The number of different authentication modes exposed through the UI. Use commas to separate multiple values.</td>
</tr>
<tr>
<td></td>
<td>• 1: I-Server standard authentication</td>
</tr>
<tr>
<td></td>
<td>• 8: I-Server Guest Authentication</td>
</tr>
<tr>
<td></td>
<td>• 16: I-Server LDAP authentication</td>
</tr>
<tr>
<td></td>
<td>• 64: I-Server Security Plugin Authentication (Trusted)</td>
</tr>
<tr>
<td></td>
<td>• 128: Kerberos Authentication</td>
</tr>
<tr>
<td></td>
<td>• 1048576: SAML Authentication</td>
</tr>
<tr>
<td>auth.modes.default</td>
<td>Which of the &quot;available&quot; modes (listed above) is shown first (default).</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>When Trusted authentication is selected as available or default authentication mode, a provider of this authentication must be chosen.</td>
</tr>
<tr>
<td>auth.trusted.provider</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1: Tivoli</td>
</tr>
<tr>
<td></td>
<td>• 2: SiteMinder</td>
</tr>
<tr>
<td></td>
<td>• 3: Custom</td>
</tr>
<tr>
<td></td>
<td>• 4: OracleAccessManager</td>
</tr>
<tr>
<td></td>
<td>• 6: Ping</td>
</tr>
<tr>
<td><strong>Intelligence Server Configuration</strong></td>
<td></td>
</tr>
<tr>
<td>iserver.default.hostname</td>
<td>The hostname of the Intelligence Server to connect to.</td>
</tr>
<tr>
<td></td>
<td>Default: NONE (must provide value)</td>
</tr>
</tbody>
</table>

© 2019, MicroStrategy Inc.
<table>
<thead>
<tr>
<th><strong>Property</strong></th>
<th><strong>Specifies</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>iserver.default.port</td>
<td>The port of the Intelligence Server to use. Default: NONE (must provide value)</td>
</tr>
<tr>
<td>iserver.initialPoolSize</td>
<td>The initial pool size of HTTP(S) connection threads Default: 10</td>
</tr>
<tr>
<td>iserver.maxPoolSize</td>
<td>The pool size of maximum allowed HTTP(S) connection threads Default: 100</td>
</tr>
<tr>
<td>iserver.requestTimeout</td>
<td>Socket connection timeout in milliseconds for I-Server request Default: 120000</td>
</tr>
<tr>
<td>iserver.tlsEnabled</td>
<td>Set to true if Intelligence Server is TLS enabled but with a private Root CA certificate or a self-signed certificate. In this case, a trustStore file and its password must be provided For Intelligence Server that is TLS enabled with a public Root CA certificate, set to false. Default: false</td>
</tr>
</tbody>
</table>

**Collaboration Server Configuration**

<table>
<thead>
<tr>
<th><strong>Property</strong></th>
<th><strong>Specifies</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>services.collaboration.enabled</td>
<td>Whether the Collaboration Server is enabled Default: false</td>
</tr>
<tr>
<td>services.collaboration.baseURL</td>
<td>The URL path to the Collaboration Server Default: NONE</td>
</tr>
<tr>
<td>services.collaboration.tlsEnabled</td>
<td>Set to true if Collaboration Server is TLS enabled but with a private Root CA certificate or a self-signed certificate. In this case, a trustStore file and its password must be provided For Collaboration Server that is TLS enabled with a public Root CA certificate, set to false.</td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td><strong>Specifies</strong></td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>IdentityToken Configuration</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>identityToken.secretKey</td>
<td>Allow services such as Collaboration Server to validate user session.</td>
</tr>
<tr>
<td></td>
<td>Secret key phrase (spaces allowed) for signing web token.</td>
</tr>
<tr>
<td></td>
<td>Default: NONE (must provide value)</td>
</tr>
<tr>
<td>identityToken.algorithm</td>
<td>Token Algorithm</td>
</tr>
<tr>
<td></td>
<td>Default: HS512</td>
</tr>
<tr>
<td>identityToken.timeToLiveSeconds</td>
<td>Time to live for token in seconds.</td>
</tr>
<tr>
<td></td>
<td>Default: 300</td>
</tr>
<tr>
<td></td>
<td><strong>SSL/TLS Configuration</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>trustStore.path</td>
<td>If Intelligence Server and/or Collaboration Server is TLS enabled but with a private Root CA certificate or a self-signed certificate, a trust store file is required.</td>
</tr>
<tr>
<td></td>
<td>The relative path (inside the WAR file) to the Trust Store file.</td>
</tr>
<tr>
<td></td>
<td>Also, supports Optional prefix <strong>classpath</strong>: (for a class path resource) or <strong>file</strong>: (for an absolute path):</td>
</tr>
<tr>
<td></td>
<td>• <strong>Example</strong> (Relative path):</td>
</tr>
<tr>
<td></td>
<td>trustStore.path=/WEB-INF/truststore.pfx</td>
</tr>
<tr>
<td></td>
<td>• <strong>Example</strong> (absolute path - windows):</td>
</tr>
<tr>
<td></td>
<td>trustStore.path=file:c:/tmp/truststore.pfx</td>
</tr>
<tr>
<td></td>
<td>• <strong>Example</strong> (absolute path - unix):</td>
</tr>
<tr>
<td></td>
<td>trustStore.path=file:/tmp/truststore.pfx</td>
</tr>
<tr>
<td>Property</td>
<td>Specifies</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>trustStore.path</td>
<td>re.pfx</td>
</tr>
<tr>
<td><strong>Example (class path):</strong></td>
<td></td>
</tr>
<tr>
<td>trustStore.path=classpath:config/truststore.pfx</td>
<td>Default: NONE</td>
</tr>
<tr>
<td>trustStore.passphrase</td>
<td>The passphrase used in the construction of the Trust Store file</td>
</tr>
<tr>
<td>Default: NONE</td>
<td></td>
</tr>
<tr>
<td><strong>Server Configuration</strong></td>
<td></td>
</tr>
<tr>
<td>session.workingSet</td>
<td>The number of recent reports/documents kept in memory for manipulation. Will apply to all users who connect to the server</td>
</tr>
<tr>
<td>Default: 10</td>
<td></td>
</tr>
<tr>
<td>session.searchWorkingSet</td>
<td>The maximum number of concurrent searches which stay in memory. Will apply to all users who connect to the server</td>
</tr>
<tr>
<td>Default: 3</td>
<td></td>
</tr>
<tr>
<td><strong>Web/Mobile Server Properties</strong></td>
<td></td>
</tr>
<tr>
<td>mobile.configuration.folder</td>
<td>The relative path to Mobile resources (inside the WAR file)</td>
</tr>
<tr>
<td>Default: NONE</td>
<td></td>
</tr>
<tr>
<td>customCss.excludePlugins</td>
<td>Exclude the list of custom plugin folder to not load the custom css files. This property will take a list of comma-separated customization plugin names.</td>
</tr>
<tr>
<td>security.allowAllOrigins</td>
<td>Specifies valid parents domain origin that may be using dossier embedded analytics. If set to true, all the origin will be allowed. This is not recommended. security.allowedOrigins =</td>
</tr>
<tr>
<td>Property</td>
<td>Specifies</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>security.allowedOrigins</td>
<td>Specifies valid parents domain origin that may be using dossier embedded analytics. The list of allowed origins will be appended to Content-Security-Policy=frame-ancestors. The request URLs affected by these constraints are defined in the GlobalHeaders filter of the web.xml</td>
</tr>
<tr>
<td>security.userInput.allowHtmlOnPromptDesc</td>
<td>Configures whether or not to allow HTML on the prompt tile and description.</td>
</tr>
<tr>
<td>auth.kerberos.config</td>
<td>The configuration file location.</td>
</tr>
<tr>
<td>auth.kerberos.keytab</td>
<td>The host keytab location.</td>
</tr>
<tr>
<td>auth.kerberos.principal</td>
<td>Host Service Principal Name (SPN)</td>
</tr>
<tr>
<td>auth.kerberos.isInitiator</td>
<td>{True/false} miscellaneous options</td>
</tr>
<tr>
<td>security.error.overrideIserverErrorMessage</td>
<td>Configures whether to override the unhandled iserver error message with the customErrorMessage.</td>
</tr>
</tbody>
</table>

Configuring Collaboration Server
Collaboration Server Configuration Properties

Settings for the Collaboration Server can be changed by editing the `config.json` file. All parameters are case sensitive and must be entered correctly for changes to take effect.

The `config.json` can be found in:

**Windows:** `<INSTALL_PATH>\MicroStrategy\Collaboration Server`

**Linux:** `<INSTALL_PATH>/CollaborationServer`

This file includes the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Specifies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port and URL Properties</td>
<td></td>
</tr>
<tr>
<td>port</td>
<td>The communication port number</td>
</tr>
<tr>
<td>dburl</td>
<td>The MongoDB URL</td>
</tr>
</tbody>
</table>

Format:

```
mongodb://{USER_NAME}:{PASSWORD}@{MACHINE_NAME}:{PORT}/{DB_NAME}?authSource={AUTH_DB_NAME}
```

where:

- `{USER_NAME}` = the user name to use when accessing the MongoDB database in a secure fashion
- `{PASSWORD}` = the password for the above user
- `{AUTH_DB_NAME}` = the database which houses the user credential information for authentication purpose

If the MongoDB database is not secured (and no credentials are required), omit the string `{USER_NAME}:{PASSWORD}@` and the string `?authSource=admin`.

- `{MACHINE_NAME}` = the fully qualified DNS name (or IP address)
<table>
<thead>
<tr>
<th>Property</th>
<th>Specifies</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORT</td>
<td>{PORT} = the port to use when connecting to MongoDB Server</td>
</tr>
<tr>
<td></td>
<td>(Default: 27017)</td>
</tr>
<tr>
<td>DB_NAME</td>
<td>{DB_NAME} = the name of the MongoDB database to store</td>
</tr>
<tr>
<td></td>
<td>Collaboration Server data. If the database does not exist, then one</td>
</tr>
<tr>
<td></td>
<td>will be created</td>
</tr>
<tr>
<td></td>
<td>If SSL has been enabled: add string &amp;ssl={true</td>
</tr>
<tr>
<td></td>
<td>If Replica Set has been configured, add string &amp;replicaSet=</td>
</tr>
<tr>
<td></td>
<td>{REPLICA_SET_NAME} at the end, where {REPLICA_SET_NAME} is the name of</td>
</tr>
<tr>
<td></td>
<td>the Replica Set</td>
</tr>
<tr>
<td>authorizationServerUrl</td>
<td>The URL endpoint of the MicroStrategy Library Server. It should end</td>
</tr>
<tr>
<td></td>
<td>with &quot;/api&quot;.</td>
</tr>
<tr>
<td>TLS Support Properties</td>
<td></td>
</tr>
<tr>
<td>enableTls</td>
<td>Whether TLS support is enabled</td>
</tr>
<tr>
<td></td>
<td>If true, TLS support is enabled, assuming that either of the following</td>
</tr>
<tr>
<td></td>
<td>pairs of fields are specified:</td>
</tr>
<tr>
<td></td>
<td>• privateKeyFile and publicCertFile</td>
</tr>
<tr>
<td></td>
<td>• keystoreFile and passphrase</td>
</tr>
<tr>
<td></td>
<td>The keystoreFile and passphrase properties have a higher priority</td>
</tr>
<tr>
<td></td>
<td>than the privateKeyFile and publicCertFile fields.</td>
</tr>
<tr>
<td></td>
<td>If omitted or false, then TLS support is disabled.</td>
</tr>
<tr>
<td>privateKeyFile</td>
<td>The path to the private key.</td>
</tr>
<tr>
<td>publicCertFile</td>
<td>The path to the public certificate.</td>
</tr>
<tr>
<td>keystoreFile</td>
<td>The path to the PKCS12 or PFX keystore file.</td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td><strong>Specifies</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>passphrase</td>
<td>The passphrase used in the Certificate Signing Request (CSR)</td>
</tr>
<tr>
<td>trustedCert</td>
<td>An array of paths (relative to the current directory) of trusted certificate files.</td>
</tr>
<tr>
<td></td>
<td>This is only needed when Library Server is in HTTPS but with a private Root CA certificate or a self-signed certificate.</td>
</tr>
<tr>
<td></td>
<td>We do not need all certificates in the chain, but only the Root CA certificate.</td>
</tr>
<tr>
<td></td>
<td>Each Root CA certificate needs to be in its own PEM file.</td>
</tr>
</tbody>
</table>

**Clustering Properties**

| **scaling**       | The type of scaling to use. Values include "horizontal", "vertical", and "none". If omitted scaling defaults to "vertical", if "cpus" is greater than 1. Otherwise, it defaults to "none". |
| redisServerUrl    | The URL of the Collaboration Cluster Cache (only supported with Redis Server). Only required if scaling is horizontal.                          |
| cpus              | The maximum number of worker processes to create on this machine. A number greater than 0 specifies the number of worker processes to create. A value of -1 indicates to create one worker process for each core on the machine. Defaults to 1. |

**User and Group Cache Properties**

| **resetCacheInterval** | A non-negative integer which specifies the interval (in minutes) to reset the users and groups information cache. Using a value of 0 applies the default time of 1440 minutes (24 hours). |

**General Configuration Properties**

| **logging**         | Whether diagnostic logging is enabled for the Collaboration Server.                                                                           |
| socketIo            | An object which holds socket.io settings. Can set pingInterval and pingTimeout. \{ "pingInterval": 25000, "pingTimeout": 30000 \} |

**Library Administration Support Properties**

| **secretKey**       | To establish trusted communication between Library Server and |
If the MicroStrategy Collaboration Server is installed separately from the MicroStrategy Library, you need to manually configure the connection information by modifying configuration property file config.json. See the appropriate Windows or Linux section for more information.

By default, the Library is accessed using the following URL and communication port:

http://localhost:8080/MicroStrategyLibrary

By default, the Collaboration Server is accessed using the following URL and communication port:

http://localhost:3000

To check the status of the Collaboration Server, you can visit this URL:

http://localhost:3000/status

This only works from the machine where the MicroStrategy Collaboration Server is installed. To access it from another machine, replace "localhost" with the fully qualified domain name (FQDN) of the machine.

The communication port must be available for Collaboration Server to be started.

The communication between components is HTTP, not HTTPS. This deployment is useful for initial testing in a lab environment, but not
recommended for production. For a production environment, where security is a priority, it is recommended to implement a secure deployment.

Manually Configure Collaboration Server to Library

You may need to install the Collaboration Server on a machine other than where the Library Server is installed. Once the Collaboration Server is installed separately, administrators need to manually configure it.

How to Manually Configure Collaboration Server to Library

1. Locate the config.json file and open it in a text editor.

```
{MSTR_INSTALL_DIRECTORY}\CollaborationServer
{MSTR_INSTALL_DIRECTORY}/CollaborationServer
```

2. Edit the file and enter the required fields:

```
{
"port": <Collaboration Server port number>
"dburl": "mongodb://{USER_NAME}:{PASSWORD}@{MACHINE_NAME}:${PORT}/
{DB_NAME}?authSource={AUTH_DB_NAME}"
"authorizationServerUrl": "{PROTOCOL}:://{LIBRARY_HOSTNAME}:{LIBRARY_PORT}/MicroStrategyLibrary/api"
}
```

- `<Collaboration Server port number>` is your Collaboration Server port number.
- `{USER_NAME}` is the user name to access the MongoDB database.
- `{PASSWORD}` is the password for the above user.
- `{MACHINE_NAME}` is the fully qualified DNS name (or IP address) of the machine running MongoDB Server.
• `{PORT}` is the port to use when connecting to MongoDB Server (Default: 27017).

• `{DB_NAME}` is the name of the MongoDB database to store Collaboration Server data. If the database does not exist, one will be created.

• `{AUTH_DB_NAME}` is the database that houses user credentials for authentication purposes.

• `{PROTOCOL}` is the web protocol that is used. http or https.

• `{LIBRARY_HOSTNAME}` is the fully qualified DNS name (or IP address) of the machine running Library Server.

• `{LIBRARY_PORT}` is the port that the Library Server is running on.

3. Restart Collaboration Server using the command ./collaborationServer.sh restart.

   To check the status of the Collaboration Server, visit http://{MONGODB_HOSTNAME}:3000/status.


5. Select **Collaboration Server** from the left control panel.

6. Enter the following information:

   • **Collaboration Server Machine Information**: Enter your Server URL.

   • **Collaboration Server Settings**: [Optional] Select Enable Logging or Enable TLS.

   • **Trusted Certificate Setting**: [Optional] Enter as many trusted certificate paths as needed.
- **Collaboration Store Settings**: Enter your **Machine**, **Port**, **Authentication database**, and **Comments database**.

- **Scaling Setting**: By default, **None** is selected. Edit as needed.

**Collaboration Server Command Line Administration Tool**

MicroStrategy Collaboration Server ships with its own command line administration tool called `admintool.js`. This administration tool allows users to search, view, and delete information in the Collaboration Server repository without directly interacting with the MongoDB database or constructing MongoDB queries.

To invoke the Collaboration Server admin tool, you need to run the Node.js executable and supply the file path for the admin tool and the Collaboration Server configuration file.

The Node.js executable (**node.exe**) can be found here:

- **Windows**: `<INSTALL_PATH>\Common Files\MicroStrategy\nodejs\node.exe`

- **Linux**: `<INSTALL_PATH>/NodeJS/<NODEJS_LINUX_VERSION>/bin/node`

The **admintool.js** file can be found here:

- **Windows**: `<INSTALL_PATH>\MicroStrategy\Collaboration Server\node_modules\mstr-collab-svc\admintool.js`

- **Linux**: `<INSTALL_PATH>/CollaborationServer/node_modules/mstr-collab-svc/admintool.js`

The Collaboration Server configuration file (**config.json**) can be found here:

- **Windows**: `<INSTALL_PATH>\MicroStrategy\Collaboration Server\config.json`
Linux: `<INSTALL_PATH>/CollaborationServer/config.json`

Starting the Administration Tool

1. Open a terminal application.
2. Enter the three file locations in the following format:

   ```
   $ "<PATH_TO_NODE>" "<PATH_TO_ADMIN_TOOL_FILE>" "<PATH_TO_CONFIG_FILE>"
   ```

   You can add the Node.js executable to your PATH variable and change to the Collaboration Server directory. If so, you can enter:

   ```
   $ node node_modules/mstr-collab-svc/admintool.js config.json
   ```

   The welcome page is shown in console after the tool starts.

   ```
   Welcome to the Collaboration Server Administration
   Type "help" to list the available commands.
   cmd>
   ```

Command List

The Admin Tool recognizes the following commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>Shows help page.</td>
</tr>
<tr>
<td>quit</td>
<td>Exits the tool.</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td><strong>Result</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td><strong>exit</strong></td>
<td>Exits the tool.</td>
</tr>
<tr>
<td><strong>show summary</strong></td>
<td>Shows a summary of the database. Shows information about topic with a specific ID. This ID uniquely identifies the topic. You can find the ID by using the listing of a topic using one of the other commands.</td>
</tr>
<tr>
<td><strong>show topic &lt;ID&gt;</strong></td>
<td>Shows information about topic with a specific ID. This ID uniquely identifies the topic. You can find the ID by using the listing of a topic using one of the other commands.</td>
</tr>
<tr>
<td><strong>show topics all</strong></td>
<td>Shows information about all topics.</td>
</tr>
<tr>
<td><strong>show topics empty</strong></td>
<td>Shows information about topics that have no comments associated with them.</td>
</tr>
<tr>
<td><strong>show topics non-empty</strong></td>
<td>Shows information about topics that have comments associated with them.</td>
</tr>
<tr>
<td><strong>show comments for-topic &lt;ID&gt;</strong></td>
<td>Shows comments</td>
</tr>
<tr>
<td>Command</td>
<td>Result</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>show comments for-user &lt;ID&gt;</td>
<td>Shows comments made by specific user. The ID refers to the MicroStrategy User DSS ID stored in MicroStrategy metadata.</td>
</tr>
<tr>
<td>show comments for-dossier &lt;ID&gt;</td>
<td>Shows comments associated with a specific dossier (base document) ID stored in MicroStrategy metadata.</td>
</tr>
<tr>
<td>show comments older-than-date &lt;ISO_DATE&gt;</td>
<td>Shows comments created before the specified ISO 8601 date. YYYY-MM-DDThh:mm:ssTZ.</td>
</tr>
<tr>
<td>drop topics empty</td>
<td>Drops all the topics that have no comments associated with them.</td>
</tr>
<tr>
<td>drop comments all</td>
<td>Drops all comments.</td>
</tr>
<tr>
<td>drop comments for-topic &lt;ID&gt;</td>
<td>Drops comments for a specific topic.</td>
</tr>
</tbody>
</table>
### Command

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>drop comments for-user &lt;ID&gt;</td>
<td>Drops comments for a specific user.</td>
</tr>
<tr>
<td>drop comments for-dossier &lt;ID&gt;</td>
<td>Drops comments associated with a specific dossier (base document).</td>
</tr>
<tr>
<td>drop comments older-than-date &lt;ISO_DATE&gt;</td>
<td>Drops comments created before the specified ISO 8601 date</td>
</tr>
</tbody>
</table>

The stored comments are organized into separate topics for each dossier page. Use the `for-topic <ID>` commands work with pages and the `for-dossier <ID>` commands to work on the entire document.

You can combine the conditions for `show comments` and `drop comments`. When multiple conditions are used, **all conditions must be true** to select the correct comments.

For example, `drop comments for-user <UID> for-topic T older-than-date 2018-01-01` will only drop comments that meet all the three conditions.

There is no confirmation message for the `drop command`. Always show the comments with the condition specified before you drop them to avoid errors.
Customizing Collaboration Server Notification

To customize Collaboration Server notifications, the Administrator needs the following files:

- `config.json`

  *This `config.json` is not the Application-level configuration file.*

- `email-Invite.template`

- `email-Mention.template`

By default, they are located in these folders:

- **Windows:** `<INSTALL_PATH>`\MicroStrategy\Collaboration Server\node_modules\mstr-collab-svc\pluginConfig\dossier

- **Linux:** `<INSTALL_PATH>`/Collaboration Server/node_modules/mstr-collab-svc/pluginConfig/dossier

Out-of-the-Box Email Templates

Two Out-of-box templates are provided for **User Invite** (`email-Invite.template`) and **User Mention** (`email-Mention.template`) emails. They are written in HTML and CSS. The Administrator can modify any item in the template or develop their own template based on these.

Configure Notifications – config.json

The `config.json` file is where Administrator can customize Collaboration Server Notifications. Administrator can enable or disable notifications, specify what email templates to use, modify the email subject and the device type for push notification.
Changing a Communication Port

Follow the instructions below to change a communication port:
• **Collaboration Server**: To change the communication port for the Collaboration Server, change the port property in the `config.json` file and restart the Collaboration Server.

  ![](image)

  All parameters in the `config.json` file are case sensitive.

• **Collaboration Server repository**: To change the communication port for the MongoDB, change the port in the `dburl` field of the `config.json` file and restart the Collaboration Server.

  The `config.json` file can be found in the following directory:

  - **Windows**: `<INSTALL_PATH>\MicroStrategy\Collaboration Server`
  - **Linux**: `<INSTALL_PATH>/CollaborationServer/config.json`

  ![](image)

  Be sure to open the Collaboration Server port in the firewall if you are employing a firewall.

**Backing Up Collaboration Store**

The Collaboration Server utilizes MongoDB as its repository database. MongoDB has a built-in command for database backup, transfer, and restoration.

MicroStrategy recommends that the MongoDB database be backed up on a regular basis.

![](image)

Here the following information from the `dburl` field in the `config.json` file from each server:

- Username
- Password
- Collaboration database name
Admin database name

1. Stop the Collaboration Server and the MongoDB service.
2. Execute the following command from the destination server:
   
   ```
   mongodump --host <source_server_name> --db <source_db_name> --username <source_username> --password <source_password> --authenticationDatabase admin --excludeCollection sessions --archive |
   mongorestore --username <target_username> --password<target_password> --authenticationDatabase admin --drop --archive
   ```
3. Restart the Collaboration Service and the MongoDB service.

How to Set Up Application Servers

The sections below provide instructions for setting up different application servers to deploy MicroStrategy Library.

Deploy MicroStrategy Library on WebSphere

The steps below apply to WebSphere version 9.

Disable WebSphere’s built-in JAX-RS.

1. In the IBM WebSphere Application Server, expand Servers > Server Types > WebSphere Application Servers.
2. Under Application servers, select <server name>.
4. Add a new property.
   
   - **Name:**
     
     com.ibm.websphere.jaxrs.server.DisableIBMJAXRS Engine
   
   - **Value:** true

Add Properties to the Web Container Settings

1. Choose **Servers > Server Types > WebSphere application servers** from the WebSphere main configuration page.

2. Click [YOUR SERVER].

3. Under **Configuration**, choose **Container Settings > Web Container Settings > Web Container**.

4. Choose **Additional Properties > Custom properties**.

5. Click **New**.

6. Add the following fields and click **OK**.

   - **Name:**
     
     com.ibm.ws.webcontainer.emptyServletMappings
   
   - **Value:** true

7. Click **OK**.

8. Click **Save to the master configuration**.

Add a Specific Web Container Custom Property

1. Choose **Servers > Server Types > Application Servers**.

2. Select the server used for Library.

3. Choose **Web Container Settings > Web Container**.
4. Click **Custom properties**.
5. Click **New**.
6. Enter the property values listed in the following table.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>com.ibm.ws.webcontainer.invokeFlushAfterService</td>
</tr>
<tr>
<td>Value</td>
<td>false</td>
</tr>
<tr>
<td>Description</td>
<td>See the corresponding IBM support page</td>
</tr>
</tbody>
</table>

7. Click **OK**.
8. Click **Save**.
9. Restart your application server.

**Install and Configure the War File**

1. In the **Select installation options** screen of the WebSphere installation process, make sure to enter the correct path in the **Directory to install application** textbox and then click **Next**.

2. On the **Map virtual hosts for Web modules screen**, select the checkbox next to MicroStrategy Library and click **Next**.

3. On the **Map modules to servers screen**, select the checkbox next to your MicroStrategy Library installation and click **Next**.

4. On the **Map context roots for Web modules page**, enter `/MicroStrategyLibrary` as the Context Root then click **Next**.

5. Select the **metadata-complete attribute** checkbox next to your MicroStrategy Library installation on the metadata for modules page and click **Next**.
6. Click Finish and make sure to Save directly to the master configuration.

7. Click the Manage Applications link
   or
   Choose Applications > WebSphere Enterprise Applications.

8. Click Library War.

9. In the Detail Properties section, click Class loading and update detection.

10. Under Class Loader Order, select Classes loaded with local class loader first (parent last).

11. Click OK.

12. In the Modules section, click Manage Modules.

13. Click Library Module.

14. In the Class Loader Order pane, select Classes loaded with local class loader first (parent last).

15. Click OK twice.

16. Click Save.

17. Choose Select check box for MicroStrategyLibrary_war.

18. Click Start.

Review the Server Class Loader Policy

1. Choose Server > Server Types > Application Servers > Your server name.

2. Ensure the Server-specific Application Settings are in one of the following orders:
Class loader policy is set to **Multiple**.

Class loader policy is set to **Single** and the class loading mode is set to **parent-last**.

Class loader policy is set to **Multiple** and the class loading mode of all applications, other than Library applications, is set to **parent-first**.

**Deploy MicroStrategy Library on JBoss**

**Before Deploying the MicroStrategyLibrary.war File:**

Open /jboss-eap-7.0/standalone/configuration/standalone.xml and search for "jaxrs" and disable following lines:

```xml
<!--extension module="org.jboss.as.jaxrs"-->
<!--subsystem xmlns="urn:jboss:domain:jaxrs:1.0"-->
```

**Deploy by Exploded Folder:**

1. **Unzip** MicroStrategyLibrary.war file as a folder, the folder name is MicroStrategyLibrary.war.

2. **Put the** MicroStrategyLibrary.war folder under ./jboss-eap-7.0/standalone/deployments/.

3. **Download** modules_config.zip and **extract it under the** jboss-eap-7.0/modules folder.

4. **Download** jboss-deployment-structure.xml and **put it under** jboss-eap-7.0/standalone/deployments/MicroStrategyLibrary.war/WEB-INF/.
5. In the ./jboss-eap-7.0/standalone/deployments/folder create a marker file named
   MicroStrategyLibrary.war.dodeploy.

6. After deploying the marker file would be updated to
   MicroStrategyLibrary.war.deployed, if we want to
   redeploy the folder, just rename it as before.

7. Open MicroStrategyLibrary/admin page to configure
   connection to Intelligence Server and Collaboration Server.

Deploy MicroStrategy Library on Wildfly

Before Deploying the MicroStrategyLibrary.war File:

1. Change directories to the folder where you want to unzip the
   WAR file.

2. Run 'jar xvf xxxx.war' to unzip the WAR file.

3. Modify the configOverride.properties file with
   Intelligence Server and Collaboration Server information.

4. Use 'jar -cvf xxxx.war *' to make the WAR file.

5. Place the new WAR file inside the folder.

Deploy the War File:

1. Stop Wildfly.

2. Put the MicroStrategyLibrary.war file under
   ..\wildfly-10.1.0.Final\standalone\deployments
   folder.

3. Restart Wildfly.
Deploy MicroStrategy Library on Jetty

Please be aware that Jetty 9 is compatible on Java 8, not compatible on Java 9

Before deploying the .war file security configuration of Jetty is necessary:

1. In the file [jetty_path]/etc/jetty.xml, between <Configure> markers the file should have <Call name="addBean"> markers which should read:

   <Call name="addBean"> <Arg> <New class="org.eclipse.jetty.security.HashLoginService"> <Set name="name">Administrator Authentication Area</Set> <Set name="config"><SystemProperty name="jetty.home" default="."/>/etc/realm.properties</Set> </New> </Arg> </Call>

2. In the same /etc folder add a file realm.properties containing the line:

   admin: admin,server-administrator,content-administrator,admin

   This will define a user, password, and a set of roles.

To deploy the MicroStrategyLibrary.war file:

1. Place the file into [jetty]/demo-base/webapps

2. Execute the following command:

   java -jar ../start.jar --add-to-start=jmx

3. Go to the demo-base folder and execute the following command:

   java -jar ../start.jar
Deploy MicroStrategy Library on WebLogic

War file Deployment

1. On a browser, open the WebLogic console.
2. Go to Deployments > Install.
4. Select the Install this deployment as an application option.
5. Leave all other settings on the default value and click Finish.
6. From the deployed folder, change the necessary properties (Intelligence Server hostname, Intelligence Server port, etc.)

Sample deployment folder:

C:\Oracle\Middleware\Oracle_Home\user_projects\domains\base_domain\servers\AdminServer\tmp\appmergegen_1500564917141_dossier7391.war\WEB-INF\classes\authConfig.properties

7. Restart the web server.

Deployment with War file exploded

1. On the machine where WebLogic is installed, manually unzip the MicroStrategyLibrary.war file to a specific folder.
2. On a browser, open the WebLogic console.
3. Go to Deployments > Install.
4. Add the folder where the war is unzipped to the Path field.
5. Select the radio button next to <folder name> (open directory).
6. Select option **Install as an application**.

7. Select the setting to **I will make the deployment accessible at extracted folder**.

8. Keep other default deployment values and and click **Finish**.

9. After the deployment is successful it should appear as active and healthy.

   **Sample deployment result:**

   http://localhost:7001/MicroStrategyLibrary

   To access the URL, you may need to turn off firewall on the corp/labs environment

**Linking MicroStrategy Library to MicroStrategy Web**


- Make sure you have the Use Library privilege for all projects, so you can access dossiers within all projects. See **Privileges in Library Web** for more information.

1. In MicroStrategy Web, navigate to the General Project Defaults page. See the **General Project Defaults** topic in the MicroStrategy Web Administrator Help for more information.

2. Enter the MicroStrategy Library URL into **Link to MicroStrategy Library** field, formatted as follows:

   http://HOST_NAME:PORT/MicroStrategyLibrary/
This URL should be the externally facing URL that takes into consideration any networking infrastructure such as load balancers.

3. Save and apply this change to all projects on the Intelligence Server.

How to Enable Seamless Login Between Web, Library, and Workstation

Enabling seamless login allows you to navigate between MicroStrategy Web, MicroStrategy Library, and MicroStrategy Workstation without having to re-authenticate regardless of your configured authentication mode. It uses an encrypted (secret) key to securely share the session among the applications.

For new installations of MicroStrategy 2019 seamless login will be configured and active if the prerequisite components are installed on the same machine. Distributed environments and customers upgrading to version 2019 need to configure the secret key by following the steps below.

Important Considerations

The following are some points to keep in mind while configuring seamless login between Web, Library, and Workstation.

- For Web and Library configuration, use the same Intelligence Server.
Once you configure seamless login between Web and Library, it will also work in Workstation.

For collaboration to work properly, use the same secret key in config.json.

MicroStrategy Cloud Environments

To Configure Web and Library Applications for Seamless Login

2. Go to Preferences.
3. Under Project Defaults > MicroStrategy Library configuration, enter your Library URL.
   
   <FQDN>:<port>/MicroStrategyLibrary

4. Click Apply.
5. In your cloud environment's homepage, hover over MicroStrategy Web and click Configure.
6. Click Security.
7. Enter a new secret key into MicroStrategy Library configuration. A secret key should be between 5 and 30 characters long.

   Keep the key for later.
8. Click Save.
9. In your cloud environment's homepage, hover over Library, and click Configure.
10. In the Library Server tab, in the **MicroStrategy Web > Link** field, enter the MicroStrategy Web link.

11. In the **Security Settings > Secret Key** field enter the secret key from step 7.

12. Click **Save**.

**Cluster Environments**

**To Configure the Secret Key Between Web and Library**

If the secret key is not available in the `configOverride.properties` file, you can add any phrase or passcode to the parameter to be used as the secret key. A secret key should be between 5 and 30 characters long.

1. Open the Library `configOverride.properties` with a text editor.

2. Copy the token value from the `identityToken.secretKey` parameter.

3. Edit the Web `sys_defaults.properties` file and add the following entry:

   `identityTokenSecretKey=<token_value>`

4. Restart your MicroStrategy Web server to apply the changes.
To Configure Web and Library Applications for Seamless Login

1. In MicroStrategy Web open Preferences > Project Defaults.

2. Enter your MicroStrategy Library URL

3. Open the Library Administration Control Panel
   (<FQDN>:<port>/MicroStrategyLibrary/admin).

4. Open the Library Web Server tab.

5. Enter your MicroStrategy Web URL into the Link field under MicroStrategy Web.

Ensuring Images in MicroStrategy Web Work With Content in the Library

The MicroStrategy Library and MicroStrategy Web are different web applications that do not share a common images directory. As a result, documents and dossiers in MicroStrategy Web with images in them may appear incorrectly in MicroStrategy Library if the images are stored as relative paths. To ensure that any documents or dossiers that you make available in MicroStrategy Library appear as expected with images, we recommend the following:

1. When creating documents and dossiers, it is possible to embed images into the content. When you do this, you insert an image into the content via upload and the image is referenced on disk via the MicroStrategy metadata. This is the recommended way of including images in your content since it ensures that both MicroStrategy Web and Library are able to commonly access the
image. For images embedded in your content in this method, the images will work in both Web and Library and no further action is required.

2. The second way to include images in content is via an absolute path reference. An absolute path reference typically looks like this, "https://www.microstrategy.com/getmedia/3534c22d-a593-4ecc-bbec-c8c6ac3deab5/logo.jpg". Images referenced via absolute paths will also work in both Web and Library and no further action is required.

3. The third way to include images in content is via a relative path. Whenever possible, we recommend you either embed images in your content, or use absolute path references. A relative path typically looks like this, /images/logo.jpg. Relative paths point to locations that are relative to the application that they were originally created for. As a result, a relative path defined for MicroStrategy Web will not work with MicroStrategy Library. This means that documents and dossiers deployed on MicroStrategy Web with relative paths will appear with broken images. There are two ways to fix this:

- Replace all of the relative paths with embedded images or absolute paths. This is recommended when there are only a few impacted documents. However, if you have many documents and dossiers in MicroStrategy Web with many relative path images, you should use the next option.

- Copy or synchronize the /images directory from MicroStrategy Web to the /images directory in MicroStrategy Library. This will allow the Library to access the images required to render the documents correctly.
MicroStrategy Library Web Internationalization

There are 5 key aspects of internationalization which are controlled by two different settings in MicroStrategy Library Web:

**Web Browser Language Setting**

- **Library Web Interface Language**: This refers to the general interface of the Library Web application.

- **Number and Date Format**: This refers to the data; specifically to the data defined as Date type or Numerical type.

- **Intelligence Server Message Language**: This refers to the Intelligence Server messages that are displayed in the Library Web application.

**MicroStrategy User Preferences**

- **Metadata Language**: This refers to the translations of the various metadata objects. These translations are saved in the metadata database.

- **Data Language**: This refers to the translations of the actual data displayed in dossiers and documents. These translations are saved in the data warehouse.

**MicroStrategy User Preferences - Language**

From Developer, the administrator can set metadata and data languages for all users at a specific project level: **Project Configuration > Language > User Preferences**.
From Developer, the administrator can also set metadata and data languages for a specific user at a specific project level: **Project Configuration > Language > User Preferences > User Language Preferences Manager > Modify.**
From Platform Web, the administrator can set all internationalization settings for a specific user at a specific project level: Web Preferences > User Preferences > General > Language.

Platform Web vs. Library Web Internationalization Settings

Platform Web internationalization settings are controlled by MicroStrategy Web User Preferences only. The Web browser
language setting will not affect the internationalization setting of Platform Web unless the metadata setting is 'Default'.

Load Balancer and Clustering Considerations for Library & Collaboration Server

When deploying MicroStrategy Library in a load balanced environment, there are several important points to consider:

1. In 2019, the MicroStrategy Collaboration Service can be clustered both horizontally and vertically. See *Cluster the MicroStrategy Collaboration Server* for instructions.

It can be deployed behind a load balancer, but it relies on a Redis Server instance to ensure proper message distribution across all of the various instances. The load balancer must utilize "sticky sessions" to ensure that clients are directed to the instance which established the initial session. It should also support the Web Socket protocol (wss:), which relies on the HTTP 1.1 Upgrade protocol.

More details on the Redis configuration are contained below.

2. The MicroStrategy Web and Library Server applications can be clustered behind a load balancer with any number of instances, however, the load balancer must utilize "sticky sessions" to ensure that clients are directed to the instance which established the initial session.

3. When deploying a cluster of MicroStrategy Library Servers the Collaboration Servers should be configured to point to the Web load balancer, rather than a specific instance, to fully utilize the cluster.
4. To fully utilize a cluster of MicroStrategy Collaboration Servers, the MicroStrategy Library Server should be configured to point to the Web load balancer rather than a specific instance.

5. Since both web browsers and mobile apps will connect to it directly, the MicroStrategy Collaboration Server port (default :3000) must be accessible and open through the load balancer.

   The Collaboration Server port can be changed from the default value by specifying it in the `config.json` file.

6. Since both web browsers and mobile apps will connect to it directly, the MicroStrategy Library Server (default :8443) must be accessible and open through the load balancer.

Deployment Topology
In the diagram, there is a single load balancer with an external facing DNS name, loadbalancer.acme.com which is used for both Library Server and Collaboration Server configuration:

- The Library Server configOverrides.properties file uses this name in the services.collaboration.baseURL property, as in: https://loadbalancer.acme.com:3000.

- The Collaboration Server config.json file uses this name in the authorizationServerUrl property, as in: https://loadbalancer.acme.com:8443/MicroStrategyLibrary/api.

In this configuration, the Collaboration Server specifies horizontal scaling by setting two properties in the config.json file:

- "scaling": "horizontal"

- "redisServerUrl":
  "redis://sharedmachine.acme.com:6379"

In the diagram, there is a shared machine used by each of the instances behind the load balancer. On this machine, there are two essential services:

- MongoDB Server - version 4.0.0 or higher.

- Redis Server - version 4.0.0 or higher.

In this environment, both shared services - MongoDB Server and Redis Server - may themselves by replicated. In this case, you must specify the correct URL in the config.json file properties:

- "dburl": "<MongoDB URL>"

- "redisServerUrl" : "<Redis URL>"
Using the Collaboration Server Outside of a Load Balanced Environment

The Collaboration Server can be run outside of a load balanced environment, on a single machine. To configure the Collaboration Server to run without any form of scaling, set "scaling": "none".

The Collaboration Server can also run multiple processes on the same machine with vertical scaling by setting the following config.json fields:

- "scaling": "vertical"
- "cpus": "<number of worker processes to launch>"

Load Balancer and Clustering for Library: Example Setup Workflow

Install and Configure Redis Server

1. Download a proper Redis version and install, following instructions from Redis Labs Ebook - Chapter Appendix.

2. Run Redis Server.

3. Check if Redis is working properly by sending a 'PING' command in redis-cli.

A 'PONG' will be responded if Redis Server is good.

```
127.0.0.1:6379> ping
PONG
127.0.0.1:6379>
```

Collaboration Servers require a URL to connect to the running Redis Server for horizontal clustering purpose.
Example: redis://redis_host_machine:6379

Modify the Redis configuration file so it can accept external traffic.

- In Redis configuration file (redis.conf or redis.windows-service.conf), add/change the bind entry to bind 0.0.0.0
- Save modification and restart Redis Server

Install and Configure MongoDB Server

1. Install MongoDB either as part of your MicroStrategy Installation (MongoDB is installed together with Collaboration Server) or do a standalone installation (Install MongoDB).

Collaboration Servers need a URL to connect to the running MongoDB Server installed.

Example: mongodb://username:password@mongodb_host_machine:27017/collab_db?authSource=auth_db

2. Modify the MongoDB configuration file so it can accept external traffic.

- In MongoDB configuration file (mongod.conf), add/modify the bindIp or bindIpAll entry

```plaintext
net:
bindIp: '0.0.0.0'
-OR-
net:
bindIpAll: true
```
Configure Collaboration Server

1. For all the Collaboration Servers installed in different machines, set Redis Server URL, web load balancer URL, clustering mode in the configuration files:

```json
{
"port": 3000,
"dburl": "<mongoDB_connection_url>",
"logging": true,
"authorizationServerUrl": "<web_load_balancer_url>/api",
"scaling": "horizontal",
"redisServerUrl": "<redis_server_url>",
........
}
```

2. Save configuration files and restart Collaboration Servers.

3. Make sure all the Collaboration Servers are running properly by visiting their individual /status URL. It will tell the actual state of the Collaboration Server Instance. For example, the Collaboration Server below is actually paused due to unreachable MongoDB Server and Redis Server.

```json
{
"version": "11.0.0.720",
"enableTls": false,
"scaling": "horizontal",
"logging": true,
"state": "paused",
}
"reason":"The following dependent services are unavailable:

"main":
{
"cpu":{"current":0,"max":17},
"memory":{"current":62758912,"max":85626880},
"connections":{"total":12,"current":4,"max":11}
"messages":{"processed":5},
"restCalls":{"received":3},
"up_since":"2018-4-18 02:12:42","pid":900
}
}

Configure Web Load Balancer Server

The load balancer must utilize "sticky sessions" to ensure that clients are directed to the instance which established the initial session. It should also support the Web Socket protocol (wss:), which relies on the HTTP 1.1 Upgrade protocol.

For example, following is the configuration for Horizontally clustered Collaboration Servers, when using NGINX as web load balancer.

```
http {

    upstream collabServer {
        ip_hash;         # this is a must, ensures that we are enable a correct socket.io handshake
        server collabServer1:3000;   # provide the list of the Collaboration
```

© 2019, MicroStrategy Inc.
Cluster the MicroStrategy Collaboration Server

To efficiently increase the number of concurrent users that collaborate in the MicroStrategy Library application, you can cluster the Collaboration Server. The MicroStrategy Collaboration Service can be clustered both **vertically** and **horizontally**, depending on your needs.

**Vertical Collaboration Server Cluster**

In a Vertical Cluster, a single machine with multiple cores hosts a single Collaboration Server which, in turn, spawns multiple processes.
to distribute the load in the server.

Cluster the collaboration vertically, you must have a machine with the MicroStrategy Library and the MicroStrategy Collaboration Server installed.

How to Cluster the Collaboration Server Vertically

1. Open the Library Administration Control Panel to verify your servers are running.

2. From the Control Panel, select Collaboration Server.

3. Under Scaling Settings, select Vertical as the scaling type.

4. From the Number of workers drop-down, select the number of CPU cores allocated to the server.

5. Click Save.

6. Restart the Collaboration Server.
Horizontal Collaboration Server Cluster

In an Horizontal Cluster of Collaboration Servers, multiple servers are hosted on multiple nodes. You can create a functional horizontal cluster configuration in several ways. One way to configure horizontally is by replicating a single working node into multiple nodes under a load balancer.

In the diagram above, Node 1 and Node 2 represent machines in a network that host the horizontal cluster of Collaboration Servers.

**Node 1** is obtained by performing a fresh installation of the MicroStrategy Platform, so it contains the MicroStrategy Library application, the Collaboration Server, and the Intelligence Server.

**Node 2** can also be obtained by performing an installation of the MicroStrategy Platform, but it is configured so the MicroStrategy Library application points to the Intelligence Server in Node 1 and the Collaboration Server points to the Collaborations Stores in Node 1. Both Nodes operate with the same data.

Lastly, both nodes are configured under a load balancer to provide clients with a single entry point to the cluster.
How to Cluster the Collaboration Server Horizontally

1. Install the MicroStrategy Platform on a single machine. This becomes Node 1.

2. In the *Library Administration Control Panel*, a warning appears after installing the MicroStrategy Platform indicating the Collaboration Server is not configured to the Library Server. Click the warning ⚠ icon and select Yes.

3. Configure the load balancer on top of Node 1. This provides external clients with a single entry point to the configuration.

4. From the Control Panel navigation, select **Collaboration Server**.

5. Under Collaboration Store Settings, modify the Machine Information to allow the Collaboration Server to utilize the Collaboration Cluster Store.

6. Under Scaling Settings, select **Horizontal** as the scaling type and enter your **Cluster Cache**.
7. Click **Save**.

8. Restart the Collaboration Server.

9. Install the MicroStrategy Platform on a new machine to introduce additional nodes in the cluster. This becomes Node 2.

10. In Node 1, open WEB-INF/classes/config/configOverride.properties and copy the values in the file. Paste the values in the configOverride.properties file in Node 2. Click **Save**.

11. In Node 1, open MicroStrategy/Collaboration Server/config.json and copy the values in the file. Paste the values in the config.json file in Node 2. Click **Save**.

12. Restart the Collaboration Server.
13. In the load balancer's configuration file, add the Collaboration Server and Library Server of Node 2. The load balancer now redirects client requests between Node 1 and Node 2.

To add more nodes to the configuration, repeat steps 9 through 13.

14. Click Save.

See Troubleshooting Your MicroStrategy Library Installation if needed.

Command Line Tools to Create Security Artifacts

There are several command line tools that assist in the creation of security artifacts, most notably, openssl and keytool. Here is how we are using these tools to construct our own security artifacts.

Creating a Key Store File Using Openssl

> openssl pkcs12 -export -out KS_FILE -inkey PRIV_FILE -in CER_FILE \ -passout file:PASS_FILE

Where:
• **KS_FILE** is the name of the key store file to create.

• **PRIV_FILE** is the path to the private key file.

• **CER_FILE** is the path to the public certificate file.

• **PASS_FILE** is the path to the file that contains the passphrase to use.

**Creating a Trust Store File Using Keytool**

```shell
> keytool -import -trustcacerts -alias collab-service \ -file CER_FILE -keystore TS_FILE -storepass PASS_PHRASE -noprompt
```

Where:

• **CER_FILE** is the path to the public certificate file.

• **TS_FILE** is the path of the trust store file to import into (or create).

• **PASS_PHRASE** is the actual pass phrase to use with the trust store.

You can create other security artifacts, such as private key, Client Signing Request (CSR) and self-signed certificate using openssl.

**Troubleshooting Your MicroStrategy Library Installation**

**Issue: My Library is Empty**

1. Open the *Library Administration Control Panel* and check that Intelligence Server is configured and running.

2. No content has been added to your Library.
3. **Monitor REST API error**

When the Library Web application behaves unexpectedly without any error message, monitor the error messages being returned from the MicroStrategy REST API using the web browser debugging tool.

1. From Chrome, open Developer Tools.
2. Open the Network tab.
3. Perform the steps to reproduce.
4. Check that all of the requests are captured.
5. Click a network request that you are interested in.
6. Within the request info window, open the Preview tab. This will display the response in JSON format.
7. Look for any network requests that are red, as this indicates that the REST API request may have failed. If you are looking for API errors only, click the XHR option so that the network tab displays all http(s) calls including the APIs.


5. Ensure you have the Use Library privilege for all projects (see **Privileges in Library Web**), so you can access dossiers within all projects.

### Issue: Collaboration Server is Unavailable

1. If the Notification Icon is not available:
   - Go to the *Library Administration Control Panel* and configure the Collaboration Server.
• Ensure you have the Use Collaboration Services privilege (see Privileges in Library Web).

2. If the Notification Icon is available but disabled, the Collaboration Server is not connected. Click the icon and use the error message dialog to contact your administrator.

3. **Monitor Web Socket error**

When Collaboration Server features within the Library fail without a detailed error message, monitor the error message directly from Web Socket messages using the web browser debugging tool.

1. From Chrome, open **Developer Tools**.
2. Open network tab.
3. Perform the steps to reproduce.
4. Filter the request type by "WS" (WebSockets).
5. Click on a network request.
6. Within the request info window, open the Frames tab. This will display the requests and responses in JSON format.

• If you don't see any request, refresh the page. It is possible that the browser debugging tool started capturing network calls after the Web Socket connection is established.

• If you don't see any requests even after refreshing, the communication between the Library Web and Collaboration Server in your environment is through HTTP polling instead of a Web Socket due to an environment limitation.
Common Configuration Errors

- The Collaboration Server is in a "paused" state because a dependent service is down or unreachable. You can see this by inspecting the status page (for example, http://localhost:3000/status) and looking at the state (value is "paused") and reason fields. The reason will explain the details, for example:
  - The following dependent services are unavailable: Authentication Server
  - The following dependent services are unavailable: Authentication Server, MongoDB Server
  - The following dependent services are unavailable: MongoDB Server
  - The "Authentication Server" refers to the "authorizationServerUrl" property in the Collaboration Server config.json file. Can you enter this URL into a browser and see a JSON response?
  - If the Tomcat service is not running, then it needs to be started. For the Collaboration Server, the "authorizationServerUrl" refers to the MicroStrategy Library Server.
  - The "MongoDB" service is defined in the config.json file by the dburl field. Are the machine and port correct? Are the user name and password values correct?
  - Error indicating that the connection is refused, such as:
    - Unable to invoke Authentication Server Method. Error: connect ECONNREFUSED 127.0.0.1:8443 (code=ECONNREFUSED)
    - Unable to connect to the Authentication Server.
    - Verify that the "authorizationServerUrl" uses the same machine name as the "Common Name" associated with your certificate. In
short, you cannot open a secure connection to a machine if you
don't have the certificate for that same machine and the certificate
is not signed by a trusted certificate authority.

- Error indicating a problem with the certificate, such as:
  - Unable to invoke Authentication Server method. Error: unable to
    verify the first certificate <code=UNABLE_TO_VERIFY_LEAF_ SIGNATURE>. Unable to connect to the Authentication Server.
  - Verify that the certificates specified in the Collaboration Server
    config.json file are the same used to create the keystore file
    used with Tomcat.

Diagnose Collaboration Server Connection Issues

MicroStrategy installations provide a command line diagnostic tool for
troubleshooting Collaboration Server connection issues.

To invoke diagnostic tool:

**Windows:**

Change the current working directory to `<MicroStrategy_install_ path>\node.js`

Run the command node "<MicroStrategy_installed_path>\Collaboration Server\node_modules\mstr-collab-svc\diagnose.js" "<MicroStrategy_installed_path>\Collaboration Server\config.json".

**Linux:**

Run the following command from the
/opt/mstr/MicroStrategy/install/CollaborationServer
directory to launch the Collaboration Server's diagnostic tool:

`. /collaborationServer.sh diagnose`
Invoking the diagnostic tool on a running system will return the following:

- Both dependent servers will show as Connected

- Collaboration server shows as Started and in a running state.

<table>
<thead>
<tr>
<th>Collaboration Server Diagnostics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependencies:</td>
</tr>
<tr>
<td>√ Authorization Server: <strong>Connected</strong></td>
</tr>
<tr>
<td>√ MongoDB Server: <strong>Connected</strong></td>
</tr>
<tr>
<td>Collaboration Server: <strong>Started on port 3000 (state=running)</strong></td>
</tr>
</tbody>
</table>

- A list of trusted certificates with the name and expiration date (only if trusted certificates are set in `config.json` file)

<table>
<thead>
<tr>
<th>Collaboration Server Diagnostics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependencies:</td>
</tr>
<tr>
<td>√ Authorization Server: <strong>Connected</strong></td>
</tr>
<tr>
<td>√ MongoDB Server: <strong>Connected</strong></td>
</tr>
<tr>
<td>Collaboration Server: <strong>Started on port 3000 (state=running)</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trusted CA Certificate Files: [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1: mstrRootCA.cer: Success: 1 Certificate</td>
</tr>
<tr>
<td>+ 'MicroStrategy Root CA' is valid from: 4-Dec-2008 =&gt; 4-Dec-2018</td>
</tr>
</tbody>
</table>

Common Errors from the Diagnostic Tool

**Collaboration Server Shows as Stopped**

| Collaboration Server Diagnostics |
Dependencies:

- Authorization Server: Connected
- MongoDB Server: Connected
- Collaboration Server: Stopped

This means the Collaboration Server is not running. You need to start the Collaboration Server and re-run the diagnostic tool.

Collaboration Server State Shows as Paused: Authorization Server Not Connected

Authorization Server hostname is found but hostname:port is not accessible. Check that port number is correct.

Collaboration Server Diagnostics

Dependencies:

- Authorization Server: Not Connected Unable to invoke Auth Server
- ! MongoDB Server: Connected
  server: [localhost:27017] on first connect [MongoError: connect ECONNREFUSED 127.0.0.1:27017]
- Collaboration Server: Started on port 3000 (state=paused)

Authorization Server hostname is found but hostname:port is not accessible. Check that port number is correct.
Authorization Server hostname:port is accessible however it's not a Library Web server. The URL being used may be incorrect, even the hostname and port number is correct.

Collaboration Server Diagnostics
Dependencies:

√ Authorization Server: Not Connected Unable to invoke

! MongoDB Server: Connected

server: [localhost:27017] on first connect [MongoDB]

Collaboration Server: Started on port 3000 (state=paused)

Authorization Server hostname is not a valid domain/ip address. You need to verify the URL being used for Authorization Server.

Collaboration Server Diagnostics
Dependencies:

√ Authorization Server: Not Connected Unable to invoke

! MongoDB Server: Connected

server: [localhost:27017] on first connect [MongoDB]

Collaboration Server: Started on port 3000 (state=paused)

Missing proper certificate for the Authorization Server which is running with TLS enabled using some private root CA signed certificates. You need to set proper values for the trusted certificate paths. See Collaboration Server Configuration Properties.

Collaboration Server Diagnostics
Dependencies:

√ Authorization Server: Not Connected Unable to invoke

! MongoDB Server: Connected
server: [localhost:27017] on first connect [MongoError: connect ECONNREFUSED 127.0.0.1:27017]

Collaboration Server: Started on port 3000 (state=paused)

Collaboration Server State Shows as Paused: MongoDB Server Not Connected

The hostname or port is not correct.

User name or password is wrong: Authentication fail

You are using replica set for MongoDB but the setName does not match.
Trust the CA Certificate Files

The diagonalistic tool will try to read each file path configured in config.json file as a trustedCert. There may be multiple certificates in a single file. Here are the possible results:

- Success
- Un-supported format
- Warning: Contains an expired certificate
- Invalid PEM block
- Success: 2 certificates
- Warning: Contains an expired certificate (There are 2 certificates inside this file. One is valid and one is expired)

Troubleshooting through Collaboration Server Logging

If Collaboration Server logging is enabled, the logs created by Collaboration Server will provide more detailed information including:

- Running warnings and errors
- Request activities
- Notification records

Collaboration Service and MongoDB Server Shell Scripts

The `./collaborationServer.sh` and `./mongod.sh` scripts can be used to resolve issues displayed by the diagnostic tool. Both scripts accept the following commands:

- **start**: Start the Collaboration or MongoDB server.
- **stop**: Stop the Collaboration or MongoDB server.
- **status**: Return the status of the Collaboration or MongoDB server.
- **restart**: Stop then start the Collaboration or MongoDB server.

Library Server to Collaboration Server Errors and Warnings

**Warning: Secret Keys Do Not Match**

Library Server and Collaboration Server need to be set up with the same secret key (a string with a minimum of 5 characters) in order to verify the other server's identity. If the secret keys do not match, the Library Server cannot talk to the Collaboration Server. Functionality such as updating the Collaboration Server configuration in Administration Control Panel will fail. The Library Server itself is okay to use.
Warning: Truststore is missing in Library Server

If the Collaboration Server is running with TLS enabled and is using a private root CA signed certificate, the Library Server is required to present the corresponding certificate in order to establish a secure communication. The administrator can configure a truststore file in the Library Server (see MicroStrategy Library Server Configuration Properties). Failure to reference the required certificate in the truststore will lead to this warning. With this warning active, the Library Server cannot communicate with the Collaboration Server. Functionality such as updating the Collaboration Server configuration in Administration Control Panel will fail. The Library Server itself is okay to use.

Warning: Cannot establish socket connection from browser to Collaboration Server

In some cases, the Library Server can talk to the Collaboration Server but end users (browsers) may fail to establish communication to the Collaboration Server (failure to establish a socket connection). There are three possible causes:

- The Library Server is running with TLS enabled (URL starting with 'https') while the Collaboration Server is running with TLS disabled (URL starting with 'http'). Some browsers will block any request to 'http' if the current website is using 'https'.

The Collaboration Server is running with TLS enabled but using a private root CA signed certificate. Some browsers will consider such private root CAs as untrusted and block requests to those servers. The administrator may need to add those private root CAs into the trust list.

The administrator may use a hostname like 'localhost' as the machine name for the Collaboration Server in the Library Server. That hostname is only recognized by the Library Server and not accessible from other users (client machine/browsers).
There may be two causes:

- **Cannot find IP address for that hostname**: end user can update the hosts file in system which can help system to resolve domain names to an IP address

- **The IP address is not visible to end user**: It may be a local ip address or the port is not exposed to public.

**Collaboration Server to Library Server Errors and Warnings**

**Warning**: The Collaboration Server is Not Targeted at the Current Library Server.

This is only a warning: the Collaboration Server is talking to a different Library Server. The current Library Server is okay to use. Potential issues may occur when the two Library Servers provide different data.

**Error**: The Collaboration Server is Not Targeted at the Current Library Server and the Target Library Server is Not Reachable.

For this case, the collaboration-related features in Library are not usable since the Collaboration Server is paused due to this error. The administrator may choose to use the current Library Server as the target Library Server for this Collaboration Server.
Error: Target Library Server is Not Reachable and Cannot be Set Up through Administration Control Panel

There are two cases when Admin UI can display issues about the current Collaboration Server but cannot provide any solutions for the problem:

- Secret keys do not match between the Library Server and the Collaboration Server.

- The Collaboration Server is TLS enabled using a private root CA signed certificate while the Library Server is missing the truststore for that certificate.

Both causes will lead to the Library Server being unable to communicate with the Collaboration Server and thus unable to control/update Collaboration Server settings.
Error: The Collaboration Store is Not Reachable

The current Administration Control Panel does not expose an interface to update the Collaboration Store. The administrator can manually update it through the Collaboration Server configuration.

Library Server to Intelligence Server Errors and Warnings

Error: Intelligence Server is Not Reachable

The administrator will see this error message when the Library Server is unable to connect to the current Intelligence Server.
The administrator will see this error message when the Library Server is unable to connect to the current Intelligence Server and Admin UI has succeeded in a connection test using the hostname and port of the current Intelligence Server. This is most likely caused by using an incorrect TLS setting.

Error: Minimum Version Requirement

If the Library Server has set a minimum Intelligence Server version requirement and the current Intelligence Server doesn't meet that requirement, this error message will be displayed.
Certificate Store Integration with MicroStrategy Library

Certificate Store primarily serves as a data source exposed via a REST API. This data source provides the data for display within the Certificates panel of Workstation.

Integration with MicroStrategy Library

To integrate Certificate Store with MicroStrategy Library, you can select Certificate Store when installing Library. This will ensure that the Certificate Store REST API appears to be a part of the entire MicroStrategy REST API suite.

So while it looks as if the Certificate Store REST API (/api/serviceCertificates) is served from the same web endpoint as the other APIs, in reality requests made to the /api/serviceCertificates endpoint are actually redirected to the Certificate Store Server. This redirection is accomplished by a proxy servlet configured in the Library web.xml file.

The servlet is configured by the installer and takes all URLs that match the url-pattern property, and redirects them to the URL specified in the targetUri property. If you are altering these settings to add or remove SSL support, you must update the targetUri property so that it specifies either "http" or "https".

If self-signed certificates are used then the trustAllCertificates and disableHostnameVerification properties must be set to true. If a user switches from self-signed to CA-assigned certificates, they should set both certificates to false to receive the benefits of the CA-assigned cert.

```xml
<servlet>
  <servlet-name>CertificateManagerService</servlet-name>
<servlet-class>org.mitre.dsmiley.httproxy.ProxyServlet</servlet-class>

<init-param>
    <param-name>targetUri</param-name>
    <param-value>http://localhost:5050/api/serviceCertificates</param-value>
</init-param>

<init-param>
    <param-name>log</param-name>
    <param-value>true</param-value>
</init-param>

<!-- SSL SUPPORT START - The following group of parameters are only relevant when the targetUri above uses the "https" protocol. -->

<init-param>
    <param-name>truststorePath</param-name>
    <param-value>{PATH_TO_TRUSTSTORE}</param-value>
</init-param>

<init-param>
    <param-name>truststoreType</param-name>
    <param-value>PKCS12</param-value>
</init-param>

<init-param>
    <param-name>truststorePassword</param-name>
    <param-value>{TRUSTSTORE_PASSWORD_OR_BLANK_IF_NONE}</param-value>
</init-param>

<init-param>
    <param-name>keyPassword</param-name>
    <param-value>{KEY_PASSWORD_OR_BLANK_IF_NONE}</param-value>
</init-param>
Certificate Store Configuration Properties

Settings for the Certificate Store can be changed by editing the application.properties file. All parameters are case sensitive and must be entered correctly for changes to take effect.

The application.properties can be found in:

- **Windows**: `<INSTALL_PATH>\MicroStrategy\Certificate Manager`
- **Linux**: `<INSTALL_PATH>/Certificate Manager`
This file includes the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Details</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.address</td>
<td>IP address Certificate Store will use to listen for incoming REST API requests. By default this is set to 127.0.0.1 to restrict access to processes on the same machine. To listen on all IP addresses specify &quot;0.0.0.0&quot; as the address.</td>
<td>If you modify any of these three settings, be sure to make the corresponding change to the targetUri property of the proxy servlet described above.</td>
</tr>
<tr>
<td>server.port</td>
<td>IP port Certificate Store will use to listen for incoming REST API requests.</td>
<td></td>
</tr>
<tr>
<td>server.servlet.context-path</td>
<td>Path to the Certificate Store endpoint.</td>
<td></td>
</tr>
<tr>
<td>server.ssl.key-store</td>
<td>The path to the PKCS12 or PFX keystore file, e.g. classpath:/opt/mstr/MicroStrategy/install/your-cert.crt</td>
<td></td>
</tr>
<tr>
<td>server.ssl.key-store-password</td>
<td>Passphrase for the keystore file</td>
<td></td>
</tr>
<tr>
<td>server.ssl.key-password</td>
<td>Passphrase for the key contained within the keystore</td>
<td></td>
</tr>
<tr>
<td>server.ssl.keyStoreType</td>
<td>Type of keystore, default is PKCS12</td>
<td></td>
</tr>
<tr>
<td>server.ssl.key-alias</td>
<td>Alias of key in keystore, used if keystore contains multiple keys</td>
<td>These settings are relevant only if Certificate Store's REST API is running over HTTPS.</td>
</tr>
<tr>
<td>Property</td>
<td>Details</td>
<td>Notes</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>max.certificate.retrieval.time.seconds</td>
<td>Maximum amount of time to allow for retrieval of all certificates from defined services. Default is 30 seconds which should be more than enough time for the services defined in 11.1.</td>
<td>These settings control which Consul agent is contacted for the list of defined services and how often the agent is contacted to refresh that list. The default settings should be sufficient for most installs and users should not need to change them.</td>
</tr>
<tr>
<td>consul.host</td>
<td>IP address of Consul agent</td>
<td></td>
</tr>
<tr>
<td>consul.port</td>
<td>IP port of Consul agent</td>
<td></td>
</tr>
<tr>
<td>consul.initialDelay.milliseconds</td>
<td>Delay (in milliseconds) before initial connection to Consul</td>
<td></td>
</tr>
<tr>
<td>consul.fixedRate(milliseconds)</td>
<td>Interval (in milliseconds) between each connection to Consul</td>
<td></td>
</tr>
<tr>
<td>wss.server.host</td>
<td>Socket.io IP address</td>
<td></td>
</tr>
<tr>
<td>wss.server.port</td>
<td>Socket.io IP port</td>
<td>These two settings control the Socket.io endpoint that GUls can connect to to receive async. updates regarding the existing services and the state of their certificates. Note that the firewall must allow WebSocket access to this endpoint (similar to the Socket.io connection in Collaboration Server)</td>
</tr>
</tbody>
</table>
### Property Details Notes

<table>
<thead>
<tr>
<th>Property</th>
<th>Details</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>spring.mail.host</code></td>
<td>Mail server IP address</td>
<td>These settings configure how Certificate Store communicates to an email host. The default is to use SMTP but user must provide the IP address, etc. By default this is not configured by the installer and must be manually enabled by the user.</td>
</tr>
<tr>
<td><code>spring.mail.port</code></td>
<td>Mail server IP port</td>
<td></td>
</tr>
<tr>
<td><code>spring.mail.protocol</code></td>
<td>Mail server protocol (SMTP)</td>
<td></td>
</tr>
<tr>
<td><code>spring.mail.properties.mail.smtp.timeout</code></td>
<td>Time out for connecting to mail host</td>
<td></td>
</tr>
<tr>
<td><code>spring.mail.properties.mail.smtp.writetimeout</code></td>
<td>Time out sending data to mail host</td>
<td></td>
</tr>
<tr>
<td><code>spring.mail.default-encoding</code></td>
<td>Encoding used to send data to mail host</td>
<td>Certificate Store periodically scans the certificates it finds to look for those that have expired or will expire within the next <code>{expiry.notification.days}</code> days. These settings control this certificate scanning process and defines which email address will receive the notifications. Multiple email addresses may be specified by comma-separated list, e.g.</td>
</tr>
</tbody>
</table>

`expiry.notification.days`
Considerations in Clustered Environments

The Certificate Store process normally listens for REST API requests on localhost:5050. The server is bound to the loopback interface so that only processes on the same machine may make API requests.

In a clustered environment, there is a Certificate Store process on each node. Because Certificate Store builds its data from the services registered with Consul, and Consul contains entries for all nodes, Certificate Store will contain entries for all services in the cluster and not those services on the current node. However, because some services are bound to the loopback address, some certificates will not be retrievable from other nodes in the cluster.

Other Library Considerations

If your Library installation does not have Certificate Store the proxy servlet may still be configured, however the targetUri will not point to a valid endpoint. Users with this configuration will receive a 502 Bad Gateway error (or other 5xx level HTTP errors). To resolve these
errors, you can install Certificate Store or remove the proxy servlet configuration from the Library web.xml file.
INSTALLING AND CONFIGURING MICROSTRATEGY IDENTITY
MicroStrategy Identity includes the following MicroStrategy products:

- MicroStrategy Communicator
- Platform Analytics

You can install MicroStrategy Identity as part of the entire MicroStrategy Secure Enterprise, or by itself. If you choose to install it alone, be aware that certain MicroStrategy components are required for Platform Analytics and MicroStrategy Communicator to operate properly; these components are installed in the background.

If you are upgrading an existing MicroStrategy Identity install, see the Upgrade Guide.

Overview of Install

The high-level steps to install and configure MicroStrategy Identity are:

- **Pre-Installation** - Verify that you have the required information, and that your server is configured with the necessary software.
- **Installation** - Perform the installation.
- **Post-Installation** - Configure MicroStrategy Identity for your enterprise environment.

For assistance with common issues, see *Troubleshooting Information*. If you require additional assistance, please contact Technical Support.
MicroStrategy Identity Pre-Installation Instructions

To prepare for installation, verify that you have the required information and that your server is configured with the necessary software for MicroStrategy Identity. Review the list of common prerequisites and the prerequisites for your specific environment (Windows or Linux) before installation:

Windows and Linux:

- You have an installation key.
- Setup the server with a fully qualified domain name (FQDN).
- Obtain the necessary SSL files.
- Open the required ports in the firewall. See Default Communication Ports.
- If SMTP requires SSL connection, obtain SMTP file server information: hostname, port, and user credentials for designated User Server administrator.
- You have verified that the install environment includes the necessary software. See the "System Requirements" section of the Readme. To verify the installation of your operating system.
  - Linux: Run the following command: `cat /etc/*-release`.
  - Windows: Run following command in a Command Prompt window: `winver` class="Note">

Windows Prerequisites

MySQL Open Source Components: Installation relies on MySQL open source components that are not provided in the MicroStrategy
installer, but can be downloaded during installation. Please note that if the server does not have Internet access you must manually copy the following files to the user's Downloads folder:

- **MySQL:**
  http://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.28-winx64.zip

- **MySQL Connector/ODBC 5.3.4:**
  http://dev.mysql.com/get/Downloads/Connector-ODBC/5.3/mysql-connector-odbc-5.3.4-winx64.msi

- **MySQL Connector/Java 5.1.22:**
  http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.22.zip

- **MySQL Connector/Python 2.1.3:**
  http://dev.mysql.com/get/Downloads/Connector-Python/mysql-connector-python-2.1.3-py2.7-winx64.msi

- **MySQL time zone description tables:**
  http://downloads.mysql.com/general/timezone_2015g_posix.zip

**Linux Prerequisites**

- You have Linux root user permission to complete the pre-installation steps.

- If you are configuring Platform Analytics and MicroStrategy Communicator after installing MicroStrategy Identity, you will need to install Python 2.7.

- Do not save files in the `/root/` folder path. Doing so will prevent a successful installation.
Pre-installation Steps

Disable SELinux

1. Disable SELinux by navigating to:

```
/SELinuxinstallpath/selinux/config
```

2. In the configuration file, change:

```
SELINUX=enforcing to SELINUX=disabled
```

3. Reboot the machine. The change will not be reflected until the machine is rebooted.

4. Verify that SELinux has been disabled:

```
sestatus
```

5. Restart the server to make the changes permanent.

Set Up Apache HTTP Server

To verify the version of the Apache HTTP server, enter the following command:

```
> httpd -v
```

If you do not have the Apache HTTP server installed, enter `yum install httpd`, then enter the following command to start the service:

```
> service httpd start
```

Download and Set Up Java Runtime Environment

1. Verify whether the JRE is already installed. Run the following command:

```
update-alternatives --config java
```
Server-JRE 8.x is recommended for this procedure.

2. Create a directory on your local computer where you want to install the JRE.

3. Choose either the .tar.gz or .rpm file.
   
   If using the .tar.gz file, the update-alternatives tool will need to be used to reconfigure the links to java tools. The command for this is as follows:

   ```bash
   update-alternatives --install /usr/bin/java java --slave /usr/bin/jjs jjs /usr/java/jdk1.8.0_172/jre/bin/java
   update-alternatives --install /usr/bin/java java --slave /usr/bin/keytool keytool /usr/java/jdk1.8.0_172/jre/bin/java
   update-alternatives --install /usr/bin/java java --slave /usr/bin/orbd orbd /usr/java/jdk1.8.0_172/jre/bin/java
   update-alternatives --install /usr/bin/java java --slave /usr/bin/pack200 pack200 /usr/java/jdk1.8.0_172/jre/bin/java
   update-alternatives --install /usr/bin/java java --slave /usr/bin/rmid rmid /usr/java/jdk1.8.0_172/jre/bin/java
   update-alternatives --install /usr/bin/java java --slave /usr/bin/rmiregistry rmiregistry /usr/java/jdk1.8.0_172/jre/bin/java
   update-alternatives --install /usr/bin/java java --slave /usr/bin/servertool servertool /usr/java/jdk1.8.0_172/jre/bin/java
   update-alternatives --install /usr/bin/java java --slave /usr/bin/tnameserv tnameserv /usr/java/jdk1.8.0_172/jre/bin/java
   update-alternatives --install /usr/bin/java java --slave /usr/bin/unpack200 unpack200 /usr/java/jdk1.8.0_172/jre/bin/java
   ```

4. To verify that the JRE was successfully installed, enter the following commands:

   ```bash
   > which java
   > java -version
   ```

Download and Set Up Apache Tomcat

1. Navigate to [http://tomcat.apache.org/download-80.cgi](http://tomcat.apache.org/download-80.cgi), and under Core in Binary Distributions, download the
.tar.gz file.

2. **Create the directory** `/opt/bin`, **untar** the `.tar.gz` file in `/opt/bin` to **create** `apache-tomcat-8.x.x`; **create a symbolic link** to this directory called `apache-tomcat-8`. **To do this:**

   ```bash
   mkdir -p /opt/bin; cd /opt/bin; tar -zxvf ~/apache-tomcat-8.0.52.tar.gz; ln -s /opt/bin/apache-tomcat-8.0.52 /opt/bin/apache-tomcat-8
   ```

3. **Install third-party libraries** into the Tomcat directory, and **note the location of the libraries**.
   
   a. **Open each of the following websites**, download the specified JAR files, and **install** them in the Tomcat library:

   - [http://central.maven.org/maven2/com/io7m/xom/xom/1.2.10/xom-1.2.10.jar](http://central.maven.org/maven2/com/io7m/xom/xom/1.2.10/xom-1.2.10.jar)
   - [http://central.maven.org/maven2/ch/qos/logback/logback-classic/1.1.2/logback-classic-1.1.2.jar](http://central.maven.org/maven2/ch/qos/logback/logback-classic/1.1.2/logback-classic-1.1.2.jar)
   - [http://central.maven.org/maven2/ch/qos/logback/logback-core/1.1.2/logback-core-1.1.2.jar](http://central.maven.org/maven2/ch/qos/logback/logback-core/1.1.2/logback-core-1.1.2.jar)
   - [http://central.maven.org/maven2/jcifs/jcifs/1.3.17/jcifs-1.3.17.jar](http://central.maven.org/maven2/jcifs/jcifs/1.3.17/jcifs-1.3.17.jar)
   - [http://central.maven.org/maven2/wsdl4j/wsdl4j/1.6.2/wsdl4j-1.6.2.jar](http://central.maven.org/maven2/wsdl4j/wsdl4j/1.6.2/wsdl4j-1.6.2.jar)
   - [http://central.maven.org/maven2/mysql/mysql-connector-java/5.1.46/mysql-connector-java-5.1.46.jar](http://central.maven.org/maven2/mysql/mysql-connector-java/5.1.46/mysql-connector-java-5.1.46.jar)
b. Compile and install tomcat-native libraries:

- **Install** apr-devel, openssl-devel, and gcc. **To do so:**
  
  ```bash
  yum install apr-devel, openssl-devel, and gcc
  ```

- **Untar the tomcat native source. To do so:**

  ```bash
  cd /opt/bin/apache-tomcat-8/ ; tar -zxvf bin/tomcat-native.tar.gz
  ```

- **Configure and compile the tomcat native libraries. To do so:**

  ```bash
  cd tomcat-native-x.x.x-src/native ;
  ./configure -with-java-home=/usr/java/jdk1.8.0_181/ ; make ; make install
  ```

- **Copy the libraries to the tomcat library directory and /usr/lib64. To do so:**

  ```bash
  cp /usr/local/apr/lib/libtncnative-1.*
  /opt/bin/apache-tomcat-8/lib/cp
  /usr/local/apr/lib/libtncnative-1.*
  /usr/lib64
  ```

4. **Start Tomcat:** > cd /((tomcat8))/bin.

5. **Display the Tomcat browser:** > ./startup.sh.

6. **In a web browser, navigate to** [http://localhost:8080](http://localhost:8080).

7. **Install Apache Tomcat Native Library. See Tomcat official documentation for detailed information.**

---

**Set Up the Timezone on Linux**

Set the system timezone to UTC localtime.

Enter the following commands:
Set up the Fully Qualified Domain Name

The following procedure requires you to work with your IT department to change the hostname. Additionally, you must obtain a trusted vendor CA-signed certificate, as well as generate a self-signed certificate. Refer to your company's IT policies for this section:

1. Contact your IT department for information about your company policy on Fully Qualified Domain Names (FQDN).

2. Change the Linux hostname to a FQDN, per your company's policy. Then have the new FQDN and IP address mapped.

3. Follow the instruction of your IT department to obtain a CA-signed MicroStrategy Identity Server SSL/HTTPS certificate from a trusted vendor.

If you are asked for a certificate request that contains your company information, perform the following steps:

1. Create and submit a certificate request (*.csr) and private key (*.key) to the vendor in the form requested.

2. Provide your company information to the certificate vendor. This information is incorporated into your certificate request.

4. Upon receiving your MicroStrategy Identity Server SSL/HTTPS Certificate and one or more Certificate Authority Certificates from the third-party vendor, create a Certificate Authority Chain (*.pem) file by combining these certificates (*.crt) files. Enter the
following command:

> cat <PathToCertFolder>/rootCA.crt
<PathToCertFolder>/intermediateCA.crt
<PathToCertFolder>/HTTPSServerCertificate.crt >
<PathToCertFolder>/UsherCACChain.pem

5. Create a self-signed signing CA certificate and append it to the CA chain. See *MicroStrategy Identity Administration*

Install memcached

Optionally, you can install memcached, which will increase the speed of transactions. After installing memcached, you must configure the modules for MicroStrategy Identity.

1. Install memcached by using either of the following methods:

   - **For yum**, use the following command: > yum install
     memcached php-pecl-memcache.

   - **For rpm**, use the following command: > rpm -i
     memcached-x.x.x-x.el6.x86_64.

2. Configure memcached by changing the following in the memcached file:

   vi /etc/sysconfig/memcached

   PORT="11211"

   USER="memcached"

   MAXCONN="1024"

   CACHESIZE="64"

   OPTIONS=""
3. Restart memcached by entering `systemctl start memcached` for RHEL 7.0+

4. Verify that the server is running by verifying the status:
   
   For RHEL 6.8 and earlier: `service memcached status`
   
   For RHEL 7.0 and later: `systemctl status memcached`

---

**Install and set up MySQL**

Create a MySQL user and grant access to each MicroStrategy product. MicroStrategy supports MySQL Community Server version 5.6.23 and above. If MySQL is already installed, continue by installing the MySQL Yum Repository.

To check the version of MySQL, connect to the MySQL command client and run the following command:

```
SHOW VARIABLES LIKE "%version%";
```

1. In a web browser, navigate to
   
   `https://dev.mysql.com/downloads/mysql`. From the drop-down list, select the correct platform of MySQL Community Server for your system. Find your operating system version, and download the RPM file.

2. Navigate to
   
   `http://dev.mysql.com/downloads/repo/yum/`, and download the correct RPM package for your system.

3. Open a command window and begin the installation by entering the following commands:

   - For MySQL yum repository, enter: `> yum localinstall mysql-community-release-el6-5.noarch.rpm`
- For MySQL Community Server, enter: > yum install mysql-community-server

- For MySQL JDBC connector, enter: > yum install mysql-connector-java

- After installation, you must set the classpath for the JDBC connector, which depends on the type of shell you are running:
  - For bourne-compatible shells, enter: > export CLASSPATH=/home/user/mysql-connector-java-5.1.34-bin.jar:$CLASSPATH
  - For C shell, enter: > setenv CLASSPATH /home/user/mysql-connector-java-5.1.34-bin.jar:$CLASSPATH

- For MySQL ODBC connector, enter: > yum install mysql-connector-odbc

- For MySQL python library, enter: > yum install mysql-python

- For MySQL PDO extension, enter: > yum install with-pdo-mysql

4. Enter: > mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -h127.0.0.1 -P3306 -uroot -p mysql to install the convert_tz functions.

If an error message displays indicating that the function is unable to load, verify that the function has been properly installed. To do this, enter the following command: > select convert_tz (current_timestamp, 'utc','est'). If the file is timestamped, then it is installed.
5. To complete the installation process, restart MySQL. Enter:

```
service mysqld start
```

6. Set the root password for MySQL.

- Enter the following command: `mysql_secure_installation`
- Enter: `mysql -u root -p`, followed by a password of your choice. Be sure to note the password, as it is needed in later steps.

7. Create the MySQL user account that you will use to grant privileges to the MicroStrategy products that you want to install:

- To create the mstr user, enter: `create user 'username'@'localhost' identified by 'password';`
- To grant the mstr user access, enter: `grant all on *.* to 'username'@'localhost' with grant option;`

8. If you intend to use MicroStrategy Identity with MySQL 5.7.4 and above, the following line needs to be added or modified in the MySQL `my.cnf` file:

```
sql-mode="STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION".
```

9. If you intend to implement MicroStrategy Identity Analytics, install crontab for MySQL:

- Verify that this directory exists: `mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -h127.0.0.1 -P3306 -umstr -p`
• Begin the installation by entering: `> yum install crontab`

• After the installation is complete, restart crontab by entering: `service crond start`

**MicroStrategy Identity Installation Instructions**

After completing the pre-installation steps, you are ready to begin the installation.

For installation instructions, see the following:

• **Windows**: *Installing MicroStrategy on Windows*

• **Linux**: *Installing MicroStrategy on Linux*

Be sure to perform all prerequisites and note MicroStrategy Identity-specific information in the installation instructions.

After completing the installation, you must perform the *MicroStrategy Identity Post-Installation Instructions*.

**MicroStrategy Identity Post-Installation Instructions**

After you have installed MicroStrategy Identity, you must deploy and complete the configuration of the MicroStrategy Identity Server and Identity Manager.

To deploy your MicroStrategy Identity Server (Linux)

1. In a command window, navigate to the installation location of the MicroStrategy Identity Server:
> cd
/
inсталлationпath
/Usher/UsherServer/usherApps/shardIDM/bin

2. Start Tomcat by entering the following command:

> ./tomcat.sh start

After starting Tomcat, the ROOT.war folder should be extracted automatically under:

<
serverпath
>/Usher/UsherServer/usherApps/shardIDM/webapps

If Tomcat does not start, see the catalina.out log located at
<
serverпath
>/Usher/UsherServer/usherApps/shardIDM/logs to view details of the problem.

3. Copy the logback classic 1.1.2 JAR file from the Tomcat directory to your MicroStrategy Identity Server root. To do this, enter the following command:

> cp /<localпath>/apache-tomcat-8.0.26/lib/logback-classic-1.1.2.jar
<serverпath>
>/Usher/UsherServer/usherApps/shardIDM/webapps/ROOT/WEB-INF/lib

4. Restart the MicroStrategy Identity Server to make the changes permanent. Enter the following commands:
5. Verify that the MicroStrategy Identity Server has restarted. In a web browser, navigate to:

https://<FQDN>:<1-way port>

where

<FQDN> is your domain name

<1-way port> is your SSL one-way port number (1443 is the default one-way port)

6. After the server is deployed, navigate back to the MicroStrategy Identity Configuration tool. Fill out the required fields with your company information to receive the activation key for your server.

To enable HTTPS for MicroStrategy Identity Manager (Linux)

Enabling HTTPS for MicroStrategy Identity Manager is optional. If you elect to enable HTTPS, then you must also enable SELinux.

To enable HTTPS

1. Install ssl_module by entering the following command:

   > yum install mod_ssl

2. Verify that the module was enabled during installation:

   > httpd -M | grep ssl_module
3. Using Vim or any other file editor, open the Identity Manager configuration located at
/etc/httpd/conf.d/manager.usher.com.conf and update the necessary file parameters.

The file should look similar to the following:

```
Listen 443  # Verify both ports are the Identity Manager port
<VirtualHost *:443>
SSLEngine on
SSLCertificateFile
<PathToCertFolder>/HTTPSServerCertificate.crt  # Verify this path is valid
SSLCertificateKeyFile
<PathToCertFolder>/HTTPSServerCertificate.key  # Verify this path is valid
SSLCACertificateFile
<PathToCertFolder>/UsherCAChain.pem  # Add this path if not already present
SetEnvIf Usher-Agent ".*MSIE.*" nokeepalive
ssl-unclean-shutdown
Alias /networkmanager
/usher/MicroStrategy/install/Usher/UsherNetworkMgr/networkmanager
  # Verify this path is valid
<Directory /usher/MicroStrategy/install/Usher/UsherNetworkMgr/networkmanager>
  # Verify this path is valid
  AllowOverride All
  Require all granted
</Directory>
</VirtualHost>
```

4. Save and close the file.
To enable SELinux

1. Open the SELinux config file:
   `vi /etc/selinux/config`

2. Replace disabled:
   `SELINUX=disabled -> SELINUX=enforcing`

3. Restart SELinux to make the changes.

---

How to complete Configuration (Windows and Linux)

Before you begin:

- SSL port numbers from your MicroStrategy installation steps
- MicroStrategy Identity Server SSL/HTTPS Certificate (.crt) and Private Key (.key) files
- Signing Certificate (.crt) and Private Key (.key) files
- SMTP server
- Google Maps API key (if you plan to use MicroStrategy Identity Manager and Google Maps)

1. Open the MicroStrategy Identity Manager page by entering the following in a web browser:

   `http://<FQDN>:<port>/networkmanager/managesystem`

2. Enter your MicroStrategy Identity Server URL and one-way port, making sure you use your FQDN (for example, `https://<FQDN>:1443`), and click **Enter**.

   > If you enter an IP Address instead of your FQDN, you will not be able to proceed.
3. In the **License Key** field, provide your installation key (universal or MicroStrategy Identity), and click **Log In**.

4. Fill out the following fields with your company information:

- **Identity Server Host**: Your auto-populated FQDN and one-way port number. This value cannot be edited.

- **System Name**: A description of the MicroStrategy Identity Server instance as it will appear in MicroStrategy Badge.

- **Identity File Directory**: Keep the default value.

- **SSL Certificate Authority Certificate**: Your self-signed .crt certificate that was generated using OpenSSL.

- **SSL Certificate Authority Key**: Your self-signed *key that was generated using OpenSSL.

- **SSL Certificate Authority Key Password**: If you have a password for your key, select the **Required** check box, and then enter your password. If you did not assign a password, leave this field blank.

- **AES Key**: Your encryption key generated during the MicroStrategy Identity Server installation. Keep the default value.

- **SAML Certificate**: Your trusted vendor CA-signed .crt.

- **SAML Key**: The key for the trusted vendor CA-signed certificate.

- **SMTP Server**: Your company's SMTP server.

  If you do not set the SMTP server and port, MicroStrategy Identity Server cannot send emails.

- **Port**: Your company's SMTP server port.
- **SMTP Authentication**: If your server is password protected, select the Required check box. Enter your username and password in the corresponding fields.

- **SMTP Configuration**: Leave this field blank.

- **Email Sender Address**: The email address that you are using to send the badge invitations for the MicroStrategy Identity network.

- **Identity Server Host**: Keep the default URL and enter the MicroStrategy Identity Server two-way SSL Port value (default value is 2443). For example, `<FQDN>:2443`. For the Identity Server Host, `https:` is not needed.

- **Gateway Host**: Keep the default URL and enter your Agent Gateway port value (default value is 9501). For the Gateway Host, `https:` is not needed.

- **Gateway Load Balancer**: Keep the default URL and enter your Agent Gateway port value to match the values for Gateway Host.

- **Google Maps API Key**: If you are using Google Maps for your MicroStrategy Identity Network, enter the third-party key.

- **Memcached**: No node is required for the MicroStrategy Identity configuration.

- **iPad Configuration Link**: If you have already set up MicroStrategy Communicator, enter your iPad URL link. This link comes from the MicroStrategy Mobile configuration.

- **iPhone Configuration Link**: If you have already set up MicroStrategy Communicator, enter your iPhone URL link. This link comes from the MicroStrategy Mobile configuration.
- **Support Email**: The email address that will be displayed in MicroStrategy Badge and used as the default address to send support emails.

- **Support Phone Label**: The label for the support number that will be displayed in MicroStrategy Badge. The default is "Phone Number".

- **Support Phone**: The support number that will be displayed in MicroStrategy Badge.

- **Push Notification-based Capabilities**: This service is not required to configure MicroStrategy Identity.

5. Click **Next**.

6. Fill in the following fields:

   - **Identity Manager Host**: Your FQDN URL. Keep the default value.

   - **Identity Manager Path**: Keep the default value.

   - **Identity Server Host**: Your Identity Server address.

   - **Certificate Path**: Leave this field blank.

   - **Network Creation**: If you do not want users to create their own network on your Identity Server, select the **Require Authentication** check box.

   - **MicroStrategy Managed Instance**: If your system is housed in Amazon Web Services, select the **Restrict LDAP Configuration** check box.

   - **Help Page Base URL**: Leave the default value.

   - **Plugin Host Server**: Keep the default value.
• **Google Client ID:** If you have a business Google Drive account, enter your Google ID number.

• **Google Client Secret:** If you want to import users from your Google Drive account, enter their information.

• **Salesforce Client ID:** If you have an executive Salesforce account for your company, enter your Salesforce ID Number.

• **Salesforce Client Secret:** If you want to import users from your Salesforce account, enter their information.

7. To save your changes and complete the configuration, click **Done**.

8. Restart the service:

   **Windows:** From the **Start** button, choose **Administrative Tools** > **Services**. Right-click on the **Apache Tomcat 8.0 shardIDM** service and select **Restart**.

   **Linux:**
   
   ```
   cd
   /<
   serverpath
   >/Usher/UsherServer/usherApps/shardIDM/bin
   > ./tomcat.sh restart
   ```

9. Click **Create an Admin**. If you need to return to this step later, type the following into the browser:

   ```
   http<s>://<
   FQDN>:<port>/networkmanager/firstUA/create.
   ```

10. To upload a photo for the Admin account, click to select a file. Supported image formats are .png, .jpeg, and .jpg.

11. Enter your first and last name.
12. In the Email Address field, type the email address you want to use with your Admin account.

13. Click Create.

You will receive an email invitation to obtain your badge.

14. Open the email in your smartphone, then click Get My Badge.

15. In a web browser, navigate to the MicroStrategy Identity Manager home page http<s>://<FQDN>:<port>/networkmanager, and scan the QR code displayed.

After you log in, you can create security networks and add users.

To create additional administrators, see MicroStrategy Identity Administration.

To synchronize Identity with an existing IDM, see Synchronizing users from Microsoft Active Directory.

To set up your MicroStrategy Identity Gateway (Windows and Linux)

After you have configured the MicroStrategy Identity Server and Network Manager, deploy the Gateway.

For Linux:

1. In a command window, navigate to the MicroStrategy Identity Server Gateway using the following:

   cd
   /<installpath>/Usher/UsherServer/usherApps/shardGateway/bin

2. Start Tomcat:

   > ./tomcat.sh start
This deploys the `gateway.war` file.

**For Windows and Linux:**

1. Check to confirm that the gateway was deployed successfully. In a browser window, type the following command:

   ```
   https://<FQDN>:<port>/gateway/test/
   ```

2. You can conduct a message push test to verify that the gateway has started. Type a phrase in the **Message** field, and click **Send** to test.

To set up an active directory with your MicroStrategy Identity Network, configure your MicroStrategy Mobile Server, or for any other MicroStrategy Identity Manager Help, see the [MicroStrategy Identity Help](#) guide.

**Troubleshooting Information**

For MicroStrategy Identity administration processes, see [MicroStrategy Identity Administration](#).

**Installation Error Codes**

For an explanation of the error codes (0 - 24) displayed by the Linux installer, see [TN300224](#).

**Default Communication Ports**

When using MicroStrategy Identity, you need to ensure that certain ports are available. The default ports are:

- **3443** - the port used by MicroStrategy Identity Manager
- **443** - an SSL-enabled port used for client-server communication with the MicroStrategy Identity Server
2443 - an SSL-enabled port for two-way (mutual) authentication with the MicroStrategy Identity Server

9501 - an SSL-enabled port used for communication with the MicroStrategy Identity Gateway

SSL Certificate Verification

You can verify your SSL certificates using the following command:

```
openssl verify -CAfile file [path_to_your_pem] [path_to_your_cert]
```

For details, see the OpenSSL documentation.

Logs

MicroStrategy Identity components log information about their operations. When troubleshooting issues, you can review the logs for each component.

Database logs

Before the database is created, error messages related to MicroStrategy Identity Manager may be found in `usher_network_manager.log`.

When installed, the MySQL instance includes a schema `usher_server_log` which includes logs for activities performed through the MicroStrategy Identity Server. The logs for MicroStrategy Identity Manager are located in a different schema and table: `usher_network_manager.usher_network_log`.

Server-side logs

The following is a list of server-side logs and their default locations. If MicroStrategy Identity is not installed in the default location, use the folder descriptions to determine where to look.
MicroStrategy Identity automatically archives many of its server logs on a daily basis. Older logs are located in the same folder, but the name will include the date.

- **MicroStrategy Identity Server Tomcat log**
  - **Location**: MicroStrategy Identity Server Tomcat logs folder
    - **Windows**: C:\Program Files (x86)\MicroStrategy\Usher\MicroStrategy Identity Server\usherApps\shardIDM\logs\catalina.out
    - **Linux**: /opt/MicroStrategy/Usher/UsherServer/usherApps/shardIDM/logs/catalina.out

- **MicroStrategy Identity Server API log**
  - **Location**: MicroStrategy Identity Server Tomcat logs folder
    - **Windows**: C:\Program Files (x86)\MicroStrategy\Usher\MicroStrategy Identity Server\usherApps\shardIDM\logs\info.log
    - **Linux**: /opt/MicroStrategy/Usher/UsherServer/usherApps/shardIDM/logs/info.log

- **MicroStrategy Identity Gateway Tomcat log**
  - **Location**: MicroStrategy Identity Gateway Tomcat logs folder
    - **Windows**: C:\Program Files (x86)\MicroStrategy\Usher\MicroStrategy Identity Server\usherApps\shardGateway\logs\catalina.out
    - **Linux**: /opt/MicroStrategy/Usher/UsherServer/usherApps/shardGateway/logs/catalina.out
MicroStrategy Identity Gateway API log

- **Location**: Root Tomcat logs folder (Windows) or MicroStrategy Identity Gateway Tomcat logs folder (Linux)

  - **Windows**: C:\Program Files (x86)\Common Files\MicroStrategy\Tomcat\apache-tomcat-8.0.30\logs\gateway.log
  - **Linux**: /opt/MicroStrategy/Usher/UsherServer/usherApps/shardGateway/logs/gateway.log

MicroStrategy Badge logs

When an error appears in MicroStrategy Badge, you can trigger the app to create an email with the log file as an attachment.

How to report an issue with a MicroStrategy Badge log

1. Open the **Settings** tab in MicroStrategy Badge.
2. Scroll down to the **Contact Us** section and tap **Report a Problem**.
3. From the pop-up menu, tap the most relevant category.
4. In the **Let us know what happened** field, describe the issue.
5. Click **Send**.

Your smartphone's email client will automatically launch a draft email addressed to the support contact for your issue. The subject and body are pre-populated, and the app log is included as an attachment.

Once the email is sent, recipients can open the log file in a text editor to view the contents.
PHP Download

Download and set up PHP

You may encounter problems while setting up PHP. To troubleshoot error messages, see your IT department and third-party reference material.

1. Use your subscription manager to enable SCL:

   `subscription-manager repos --enable rhel-server-rhscl-6-eus-rpms`

2. Enter the following yum installation commands to install php 7.2 and modules:

   `yum install php72 php72-php php72-php-gd php72-php-mbstring`

3. Install the updates database module for MySQL:

   `yum install php54-php-mysqlnd`

4. Disable the loading of php 2.3 Apache:

   `mv /etc/httpd/conf.d/php.conf /etc/httpd/conf.d/php.conf.old`

5. Restart Apache to make the changes:

   `service httpd restart`

6. After the installation is complete, enable mod_rewrite. To do this, enter `/InstallPath/etc/httpd/conf/httpd.conf` to modify the file. The parameters are as follows:

   ```
   <Directory "/var/www/html">
   Options FollowSymLinks
   AllowOverride All
   </Directory>
   ```
7. Verify that the required PHP modules have been installed correctly.

8. Enter the following commands:
   - `php -v` to check the version of PHP.
   - `php -m` to check the modules installed:
     - curl
     - zlib
     - openssl
     - gd
     - mbstring
     - mysql
     - pdo-mysql
     - zip
     - PDO
     - exit
   - `php -r 'print_r(gd_info());'` to verify the GD version and JPEG support.

9. Execute the following command to add
   
   `/etc/httpd/conf/httpd.conf`:

   ```
   LoadModule php5_module modules/libphp5.so
   AddType application/x-httpd-php .php
   ```

10. Enter the following command s to create `test.php` in
   
   `/var/www/html:`
<?php
phpinfo();
?>

11. Restart the service to complete the changes.

12. In a web browser, navigate to http://localhost/test.php. You should see the PHP landing page, which signals that the verification is successful.
ACTIVATING YOUR INSTALLATION
After your MicroStrategy installation is complete, you have 30 days to activate your installation. Before you activate your installation you must request an Activation Code from MicroStrategy. You can complete this request when you install MicroStrategy with the MicroStrategy Installation Wizard or after the installation using MicroStrategy License Manager.

This section describes the following procedures:

- *Request an Activation Code, page 341*
- *Activate Your Installation, page 344*

For answers to commonly asked questions about server activation, see *Server Activation FAQ, page 346*.

**Request an Activation Code**

You can request an Activation Code by supplying MicroStrategy with important information related to your installation. The information you provide helps MicroStrategy understand how you plan to use MicroStrategy software. With this information MicroStrategy can provide better information and technical support for your software configuration.

On Windows, MicroStrategy products can be activated only in graphics user interface (GUI) mode, using License Manager.

MicroStrategy products can be activated on Linux, either in GUI mode or in command line mode, using License Manager. In both cases, License Manager runs and requests the same information. The main differences are in how you provide the information and navigate through the Activation Code request steps.
Request an Activation Code with License Manager

If you requested an Activation Code during installation you can skip this procedure and activate your installation by following the instructions in the next procedure, *Activate your software installation*, page 344.

1. Open MicroStrategy License Manager:

   - **Windows**: Go to **Start > Programs > MicroStrategy Tools > License Manager**. License Manager opens.

   - **Linux**: License Manager can be run in GUI mode or command line mode.

     - **GUI**: In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `bin` and type `./mstrlicmgr`, then click **Enter**. The MicroStrategy License Manager opens in GUI mode.

     - **Command line**: In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `bin` and type `./mstrlicmgr -console`, then click **Enter**. The MicroStrategy License Manager opens in command line mode.

   The steps to request an Activation Code in command line mode of License Manager vary from the steps below. Refer to the License Manager command line prompts to guide you through the steps to request an Activation Code. For more information specific to requesting an Activation Code for your installation in command line mode, refer to MicroStrategy Tech Note TN13550.
2. Select the **License Administration** tab. Under Server Activation select the **Activate Server Installation** option and click **Next**.

3. Select the **Generate Activation File and Request Activation Code** option and click **Next**.

4. Enter the characteristics of your server installation and click **Next**.

5. Enter the contact information for the person who installed the software. Make sure to correctly select whether you are an employee of the licensed company or installing the software on the licensed company’s behalf.
   
   - If you select that you are an employee of the licensed company, click **Next**. Once you complete the following step, the Activation Code is sent to the email address given; therefore it is important that the email address is valid and entered correctly.
   
   - If you select that you are not an employee of the licensed company, a contact information page is displayed after you click **Next**. Enter the contact information for the licensed company. Click **Next**. Once you complete the following step, the Activation Code is sent to the email address given; therefore it is important that the email address is valid and entered correctly.

6. Select **Yes, I want to request an Activation Code now** and click **Next**.

   An email containing the Activation Code is sent to the email address or addresses you confirmed in the steps above.
Activate Your Installation

After you have requested an Activation Code, MicroStrategy sends an email to the addresses provided in the request. This email contains the Activation Code that is necessary to complete the activation of your installation.

Activate your software installation

This procedure assumes that you have requested an Activation Code and received an email from MicroStrategy containing the Activation Code:

Windows

1. Open MicroStrategy License Manager from **Start menu > Programs > MicroStrategy Tools > License Manager**.
2. Go to the **License Administration** tab > **Activate Server Installation** and click **Next**.
3. Select the **Server Activation using Activation Code** option and enter your Activation Code in the text field. Click **Next**.
4. Click **OK**.

Linux

1. License Manager can be run in GUI mode or command line mode:
   - **GUI**: In a Linux console window, browse to **HOME_PATH** where **HOME_PATH** is the specified home directory during installation. Browse to the folder **bin** and type **./mstrlicmgr**, then click
Enter. The MicroStrategy License Manager opens in GUI mode.

- Command line: In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `bin` and type `./mstrlicmgr -console`, then click Enter. The MicroStrategy License Manager opens in command line mode.

The steps to activate your installation in command line mode of License Manager vary from the steps below.

2. Go to the License Administration tab > Activate Server Installation and click Next.

   The step above is not necessary for License Manager in command line mode.

3. Select the Server Activation using Activation Code option and enter your Activation Code in the text field. Click Next.

4. Click OK when the verification message appears.

You must restart your Intelligence Server for the activation status to update. You must also restart your Web server for the activation status to update in MicroStrategy Web.

Configuring your MicroStrategy Installation

After completing the steps to activate your installation, you can continue your setup and configuration. To help guide the rest of your installation and configuration steps, refer to the section Installation and Configuration Checklists, page 110 in Chapter 1, Planning Your Installation, for installation and configuration checklists.
Server Activation FAQ

What is MicroStrategy Server Activation?

MicroStrategy Server Activation is a licensing technology that ensures that installations of MicroStrategy server products are authentic and have been legitimately licensed. Server Activation registers each Server Installation with MicroStrategy and locks the installation to a specific machine.

Why is Server Activation necessary?

Server Activation provides benefits to both MicroStrategy and its customers:

- It ensures that the software products being used are authentic.
- It helps customers in identifying software installations to prevent over-installation.
- It improves customer service by maintaining a register of the hardware configurations used by our customers.

Does MicroStrategy adhere to Software Activation common practices?

Yes. Extensive research was performed on software activation and it was found that the Business Software Alliance (www.bsa.org) provides the most comprehensive information. MicroStrategy has closely followed these best practices.

Which products require activation?

All MicroStrategy modular and non-modular server products require server activation, including:
• Intelligence Server, Intelligence Server Module, Clustering Option, Report Services, Report Services Option, OLAP Services, OLAP Services Option, Distribution Services, MultiSource Option

• Web, Web Reporter Module, Web Analyst Option, Web Professional Option

• Mobile Server

• Narrowcast Server Delivery Engine

If more than one server product is installed on the same machine, does each server product need to be activated and deactivated separately?

No. All MicroStrategy server products installed on a single machine are grouped as a Server Installation and will be activated and deactivated as a group.

Which customers are required to activate MicroStrategy server products?

All customers who install MicroStrategy server products will need to activate their Server Installations.

Is a new CD Key needed to install products on different machines?

No, Server Activation has no impact on CD Keys. The latest CD Key sent by MicroStrategy can be used to install products on many machines just as before. The only difference now, is that each installation on a different machine will need a different Activation Code to activate the installation. Server Activation is independent of the CD Keys.
Is Server Activation required for both Named-user and CPU based licenses?

Yes. Server Activation is required for both Named-user and CPU based licenses. Server Activation is designed to track software installations regardless of licensing model. Server Activation does not manage or limit the number of servers on which server software is installed.

Does Server Activation aggregate the total number of licenses installed and prevent over-installation of products?

No. Server Activation and the licensing models are independent. However, using Server Activation information available at https://licensing.microstrategy.com, organizations can monitor the number of installations. Deactivating Server Installations that are not being used will ensure this information is up-to-date.

Does Server Activation apply to Evaluation Editions?

Yes. The Evaluation Edition must be activated within 7 days of installation.

What is the procedure for activating Server Installations?

Installing, modifying or upgrading MicroStrategy Server Installations will automatically generate an Activation XML file that contains information about the installation. This XML file, called the "Activation File" is uploaded to MicroStrategy either automatically by the installation routine or through License Manager; or by manually uploading the Activation File through a web browser via a secure web site, https://licensing.microstrategy.com. MicroStrategy then creates a machine-specific Activation Code which is emailed to the installer and to the MicroStrategy Tech Support liaisons. The
Activation Code must be manually entered into License Manager on the target Server Installation.

Can the Server Installation be automatically activated after automatically requesting an activation code?

No, the Activation Code is sent to the installer and to the MicroStrategy Tech Support contacts by email. Upon receiving the activation code, the Server Installation needs to be manually activated by entering the activation code using License Manager. For manual activations, the Activation Code can be copied from the Activation website, and pasted into License Manager.

What information is sent to MicroStrategy in the Activation XML File?

The following information is sent to the Activation XML File:

- Installation Information:
  - Installation timestamp
  - Activation ID (if the installation has previously been activated)
  - Contract information
  - CD Key used in installation
  - Installer contact details – name, address, email
  - Company contact details – name, address, email
  - Server Installation information – name, location, use

- Hardware information:
  - Unique Hardware Identifier
  - CPU Information – type, bit-size, clock speed, total quantity
- Physical Memory installed

- Operating System information:
  - Type, version, bit-size, page/swap size
  - Locale
  - Additional information – 4GT mode and hyperthreading in Windows

- MicroStrategy information:
  - Install Type – new/modify/upgrade
  - Products and version installed
  - Number of CPUs allocated for CPU licenses

- Database information:
  - Metadata database and ODBC driver
  - Data warehouse database and ODBC driver

Is the information sent to MicroStrategy secure?

Yes. The Activation XML file is sent to MicroStrategy in the following ways:

- Automatically during installation or through License Manager. This information is encrypted using a RIPEMD-160 algorithm before being sent to MicroStrategy.

- Manually in the https://licensing.microstrategy.com website. Communication with this website is conducted through Secure Socket Layer once the user has been authenticated.

Can I change the information in the Activation XML file?

The content of this file is secured with a digital signature. Contact MicroStrategy Technical Support if the content is incorrect.
What is used to lock the server to a machine?

A unique hardware identifier for each machine is used to lock an installation to that machine. Any changes to these identifiers will require reactivation of the Server Installation:

- Windows: An identifier generated from a one-way hash of the network interface card MAC address
- Linux: An identifier generated from a one-way hash of the network interface card MAC address

Is there a grace period from the time server products are installed to when it can be activated?

Yes. There is a 30 calendar day grace period from installation (7 days for Evaluation Edition) before a server installation must be activated.

What happens if the Server Installations are not activated?

If a server installation has not been activated within 30 calendar days (7 days for Evaluation Edition), the server products will not be able to be restarted.

Should the installations be automatically or manually activated and deactivated?

It is preferable to automatically send the activation information to MicroStrategy, either during installation or using License Manager. This is an easy process that should take less than one minute to complete, and ensures that the correct information is sent to MicroStrategy.
What should be done if requesting an Activation Code fails during installation?

Firstly, allow the installation to complete. The Activation Code request should then be attempted using License Manager. If this does not work, activate the Server Installation manually by visiting the Activation website at https://licensing.microstrategy.com.

What should be done if automatic request for an Activation Code does not work at all?

The Server Installation should be manually activated using License Manager on a machine that has access to the Web. The Activation XML file should be copied from the Server Installation that requires activation to this computer. If this does not work, contact MicroStrategy Technical Support.

Can the Activation Code be used on a different machine?

No. The Activation Code contains the unique identifier for a specific machine and can only be used on that machine. The server products will not be activated if the incorrect Activation Code is used.

What if the Server installation has to be moved to another machine?

If the Server Installation needs to be moved to another machine, uninstall the MicroStrategy products or deactivate the Server Installation from License Manager. Notify MicroStrategy that Server Installation has been deactivated. This keeps your active server inventory up to date as shown to you on https://licensing.microstrategy.com.
What if the server machine has a catastrophic failure and cannot be deactivated automatically or manually?

If a machine has a catastrophic failure and the server products cannot be uninstalled or deactivated, contact MicroStrategy Technical Support to update the status of this Server.

If there is more than one server product installed on a machine and one is removed, does the Server Installation need to be deactivated?

Upon removal of a server product, all remaining server products are automatically deactivated. The remaining server product(s) need to be reactivated to reflect the new product configuration on that machine. For example, if a machine contains MicroStrategy Narrowcast Server and MicroStrategy Web, and MicroStrategy Narrowcast Server is removed, MicroStrategy Web will be automatically deactivated. MicroStrategy Web will need to be reactivated.

What information can be monitored on the website?

When registered Technical Support liaisons log into the MicroStrategy Activation website, they can display a list of all Server Installations. This report lists each Server Installation along with the following information for reference:

- Installation ID
- Activation Status
- Contract ID
- Operating System
- Machine CPUs
- Last Update Type
• Last Updated by
• Last Update Date
• Products installed
• Activation and deactivation history

The Installation ID is a unique identifier for Server Installations. This ID is provided along with the Activation Code in the email received when requesting activation.

Does Server Activation apply to MicroStrategy Suite?

Yes. The MicroStrategy Suite must be activated within 30 days of installation.
CONFIGURING AND CONNECTING INTELLIGENCE SERVER
After installing MicroStrategy, you must complete a few configuration tasks. This chapter addresses the processes used to configure the databases you intend to use in your business intelligence system, as well as an installed MicroStrategy suite of products.

The MicroStrategy platform includes a Tutorial project, which is a sample data warehouse and a demonstration project you can use to learn about the various features that MicroStrategy offers. It is ready to be used and requires no additional configuration tasks. If you want to set up a new system using your own data, you must read this chapter and perform the tasks it describes.

This chapter includes the following information:

If you are configuring MicroStrategy on a Linux machine that does not have a GUI, you can perform configuration tasks with command line tools. For steps to perform configuration tasks using command line tools in Linux, see Chapter 12, Configuring MicroStrategy Using Command Line Tools.

Communicating with Databases

Establishing communication between MicroStrategy and your databases or other data sources is an essential first step in configuring MicroStrategy products for reporting and analyzing data. This section explains how MicroStrategy communicates with various data sources and the steps required to set up this communication.

ODBC (Open Database Connectivity) is a standard database access method. ODBC enables a single application to access database data, regardless of the database management system (DBMS) that stores the data. A DBMS is a collection of programs that enables you to store, modify, and extract information from a database.
MicroStrategy Intelligence Server, when used in a three- or four-tier configuration, is the application that uses ODBC to access a DBMS. ODBC drivers translate MicroStrategy Intelligence Server requests into commands that the DBMS understands. MicroStrategy Intelligence Server connects to several databases (at a minimum, the data warehouse and the metadata repository) to do its work.

Users of MicroStrategy Web can also connect to data sources using database connections. A database connection supports connecting to data sources through the use of DSNs, as well as through DSNless connections, to import and integrate data into MicroStrategy. For steps to create database connections in MicroStrategy Web, see *Creating Database Connections in Web, page 799.*

This section describes the ODBC standard for connecting to databases and creating data source names (DSNs) for the ODBC drivers that are bundled with the MicroStrategy applications.

The diagram below illustrates the three-tier metadata and data warehouse connectivity used in the MicroStrategy system.
The diagram shown above illustrates projects that connect to only one data source. However, MicroStrategy allows connection to multiple data sources in the following ways:

- With MicroStrategy MultiSource Option, a MicroStrategy project can connect to multiple relational data sources. For information on MultiSource Option, see the Project Design Guide.

- You can integrate MDX cube sources such as SAP BW, Microsoft Analysis Services, and Hyperion Essbase with your MicroStrategy projects. For information on integrating these MDX cubes sources into MicroStrategy, see the MDX Cube Reporting Guide.

This section provides information and instructions on the following tasks:
Setting up ODBC

The following information assists you in setting up ODBC between Intelligence Server and your metadata database and data warehouse.

ODBC is a standard method of communicating with database servers. Intelligence Server uses ODBC to connect to and communicate with all database servers in the system. Specifically, ODBC connects to and transfers data to and from data sources within relational databases.

ODBC permits maximum interoperability—an application can access data in diverse DBMSs through a single framework. A client application uses a database driver and a driver manager to make a connection to the data source. A data source, identified by a data source name, is the database or file accessed by a driver. Data source is another term for a logical database within a database server. A database server can contain multiple logical databases or data sources.

When setting up your MicroStrategy environment, you must create a separate connection to the data warehouse and metadata repository. This requirement is true even if both databases are accessed through the same DBMS. Further description of these two requirements is below:

- A data warehouse stores the data that users of the system must analyze to track and respond to business trends, and to facilitate forecasting and planning efforts.

- Metadata is a repository whose data associates the tables and columns of a data warehouse with user-defined attributes and facts to enable the mapping of business views, terms, and needs to the underlying database structure. Metadata can reside on the same server as the data warehouse or on a different server. It can be stored in different relational DBMSs.

A successful ODBC connection requires the following information:
• A data source name (DSN) is the name for a pointer used by a client application to find and connect to a data source. A data source is the database accessed by a driver. The information obtained through a DSN generally includes the host computer name or IP address, instance name, and database name. However, the exact information varies depending on the type of database server.

• An ODBC driver is a type of software that translates information between the client application (Intelligence Server) and the database server API. For more information on ODBC drivers and how they work with MicroStrategy, see ODBC drivers, page 360.

• A connection string stores the information required to connect to a database server. A connection string usually includes a DSN, as well as the user ID and password required to log in to the database server. This information varies depending on the particular database server. For MicroStrategy environments, a connection string is commonly provided by a database instance (see Creating a database instance, page 435).

ODBC drivers

ODBC drivers are DBMS-specific and must be installed on MicroStrategy Intelligence Server prior to creating the ODBC connection to the warehouse and metadata databases. MicroStrategy embeds and brands Progress and Magnitude ODBC drivers in the MicroStrategy platform. These drivers are certified to work with MicroStrategy products.

The purpose of an ODBC driver is to translate MicroStrategy Intelligence Server requests into commands that the DBMS understands. Users of the MicroStrategy platform can employ the MicroStrategy-branded ODBC drivers to connect MicroStrategy products to various DBMSs. For a list of the available ODBC drivers for Windows and Linux that are certified for Intelligence Server and
different DBMS types, see *Certified ODBC Drivers for MicroStrategy Intelligence Server, page 98.*

- Although it is possible to use a non-certified driver, it is strongly recommended that you contact your database vendor to obtain a certified driver if the selected driver is not certified as valid.

- MicroStrategy products include certified ODBC drivers for you to use. The *MicroStrategy Readme* lists these MicroStrategy ODBC drivers.

Default location for ODBC and driver files for Windows

MicroStrategy components require 64-bit drivers to achieve ODBC connectivity.

The ODBC driver manager and support libraries are commonly installed in the C:\WINDOWS\SYSTEM or C:\WINDOWS\SYSTEM32 directories. Refer to your third-party documentation for the locations of ODBC support and driver files.

The database-specific ODBC drivers are installed in the locations specified during the installation of the drivers. MicroStrategy-branded drivers are installed in C:\Program Files (x86)\Common Files\MicroStrategy on a 64-bit Windows environment.

Default location for ODBC and driver files for Linux

MicroStrategy components require 64-bit drivers to achieve ODBC connectivity.

The ODBC driver manager and support libraries are usually installed in INSTALL_PATH/lib

The database-specific ODBC drivers are installed in the locations specified during the installation of the drivers. MicroStrategy-branded ODBC drivers are installed in INSTALL_PATH/lib, where
**INSTALL_PATH** is the directory you specified as the Install Directory in the Install Wizard.

The MicroStrategy Connectivity Wizard lists only the MicroStrategy-branded ODBC drivers. However, this guide also provides information on how to install drivers from other vendors with MicroStrategy. For more information, see *Creating DSNs for Specific Data Sources*, page 759.

### Defining DSNs

After you install an ODBC driver, you can define one or more data sources for it. The DSN should provide a unique description of the data, for example, `Payroll_Project_Metadata` or `Payroll_Warehouse`.

The DSN is the name for a pointer used by a client application (in this case MicroStrategy) to find and connect to a data source. Multiple DSNs can point to the same data source and one DSN can be used by different applications.

You can define a data source connection with a DSN by using:

- The MicroStrategy Connectivity Wizard—configures connectivity to data sources by creating a DSN that uses a MicroStrategy-branded ODBC driver (see *Creating a DSN for a data source*, page 363).

- The Microsoft ODBC Data Source Administrator—creates a DSN for an ODBC driver that is not MicroStrategy-branded (see *Managing ODBC and data sources with Microsoft ODBC Data Source Administrator*, page 365).

- Edit the odbc.ini on Linux. For more information, please see *Configuring ODBC Parameters with odbc.ini*.

⚠️ It is strongly recommended you use the MicroStrategy Connectivity
Wizard when creating a new DSN for a MicroStrategy-branded ODBC driver. Use the Microsoft ODBC Data Source Administrator only if you intend to use a driver that is not MicroStrategy-branded.

⚠️ If you create DSNs using the Microsoft ODBC Data Source Administrator, you must create system DSNs. Otherwise, MicroStrategy interfaces will not recognize them.

Creating a DSN for a data source

If a DSN does not already exist in your empty metadata repository or the repository installed with MicroStrategy, you can add or create a new one.

The MicroStrategy Connectivity Wizard is a tool designed specifically to configure connectivity to data sources by creating a DSN that uses a MicroStrategy-branded ODBC driver.

To create a DSN

1. On Windows, log in to the system as an administrator.

2. From the Start menu, go to Programs > MicroStrategyTools > Connectivity Wizard.

On Linux:

- Using the Connectivity Wizard interface, perform the following steps:
  - In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation.
  - Browse to the folder bin and type ./mstrconnectwiz, and then press ENTER.
• From the command line, perform the following steps:

  • In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.

  • Browse to the folder `bin` and type `. /mstrconnectwiz -h`, and then press ENTER.

  • This command displays command line operation syntax and examples for different database platforms. Create your command based on the syntax and examples displayed. Once you perform your command, the DSN is created and no further action is required. For detailed steps on how to use the command line version of this tool, see *Creating a DSN for a Data Source*, page 699 in Chapter 12, *Configuring MicroStrategy Using Command Line Tools*.

3. Click **Next**.

4. Select a database driver with which to create a DSN and click **Next**.

   Only a few databases can contain metadata repositories. For details, refer to the *MicroStrategy Readme*. Only DSNs created to connect to these databases can be used to connect to metadata repositories.

5. Enter the information in the appropriate fields for connecting with the selected database driver. The information to enter varies depending on the database platform that you selected. For more information, see *Creating DSNs for Specific Data Sources*, page 759.

6. Click **Test**.

7. Enter the **User Name** and **Password** to connect to the database.
8. Click **Connect**. If the test is performed successfully, the connection with the database is established. If the test fails, verify the correct connection information with your database administrator and make any required changes to the information you provided in the previous steps.

9. Click **Close**, and then **Finish**.

If you already have an existing DSN with the same name as the one you provided, a message box appears. You have the following options:

- Select **Yes** to make sure the DSN points to the location you are expecting. This overwrites the existing DSN.
- Select **No** to save the DSN with a different name.

Repeat the above steps to create as many DSNs as you require. At a minimum, create one for your metadata and one for your warehouse.

Managing ODBC and data sources with Microsoft ODBC Data Source Administrator

The Microsoft ODBC Data Source Administrator manages database drivers and data sources on Windows. The Microsoft ODBC Data Source Administrator utility creates a log with which to trace calls to data sources and to view, create, and edit DSNs. The utility is available from Microsoft and is usually included with the purchase of an ODBC driver.

- It is strongly recommended that you use the Connectivity Wizard when creating a new DSN for a MicroStrategy-branded ODBC Driver. Use the Microsoft ODBC Data Source Administrator only if you intend to use a non-MicroStrategy driver.
- If you choose to create DSNs using the Microsoft ODBC Data Source Administrator, they must be system DSNs. Otherwise, MicroStrategy interfaces cannot recognize them.
To create a DSN using the Microsoft ODBC Data Source Administrator

1. Log in to the machine as an administrator. This gives you the ability to create a system-wide DSN, rather than a user-specific DSN.

2. In most Windows systems you can access the ODBC Data Source Administrator from the Control Panel. Refer to your third-party Microsoft documentation for steps to access the ODBC Data Source Administrator tool.

3. Click the System DSN tab.

   To view all the installed ODBC drivers, click the Drivers tab.

4. Click Add.

5. Select the desired driver and click Finish.

   We recommended that you select a MicroStrategy ODBC driver. These drivers, whose names start with MicroStrategy, were installed when you installed the MicroStrategy application on the computer.

6. Enter the information in the appropriate fields to create a data source for the selected database driver.

   The information to enter varies depending on the database platform that you selected, which is discussed in Creating DSNs for Specific Data Sources, page 759.

7. Click OK.

Testing ODBC connectivity

ODBC connectivity is one of two layers of connectivity that are listed in the next table, along with the associated connectivity testing
programs. Connectivity should be tested from the bottom up—the network layer first and then the ODBC layer.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Test with</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODBC driver</td>
<td>Test ODBC</td>
</tr>
<tr>
<td></td>
<td>mstrtestodbc or mstrtdodbc</td>
</tr>
<tr>
<td>Network</td>
<td>Simple Network Layer Testing Tool</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Ping, PING.EXE, for TCP/IP</td>
</tr>
</tbody>
</table>

The test method described above reflects the situation when the ODBC driver and the database network software are bundled. If they are not bundled, they must be configured and tested separately, using database-specific tools.

Using the DB Query Tool

The MicroStrategy DB Query Tool is available in Windows, and Linux Intelligence Server installations. It is used to test and troubleshoot connectivity to databases, create and execute SQL commands through ODBC, and run scripts.

Before you use the DB Query Tool, test the network layer with the network layer utility, PING.EXE. Consult your operating system or network system documentation for details.

To use the DB Query Tool

1. To use the DB Query Tool:
   - On Windows using the DB Query Tool interface, perform the following step:
From the Windows Start menu, go to Programs > MicroStrategy Tools > DB Query Tool.

- On Windows from the command line, perform the following steps:
  1. From the Windows Start menu, select Run.
  2. In the Open drop-down list, type cmd and click OK. A command prompt opens.
  3. Type todbcx.exe and click Enter. Prompts guide you through testing your ODBC connection from the command line and should be used in place of the steps below. For detailed steps on how to use the command line version of this tool, see Testing ODBC connectivity in Chapter 12, Configuring MicroStrategy Using Command Line Tools.

- On Linux using the DB Query Tool interface, perform the following steps:
  1. In a Linux console window, browse to HOME_PATH, where HOME_PATH is the directory that you specified as the home directory during installation.
  2. Browse to the folder bin and type ./mstrdbquerytool, then click Enter.

- On Linux from the command line, perform the following steps:
  1. In a Linux console window, browse to HOME_PATH, where HOME_PATH is the directory that you specified as the home directory during installation.
  2. Browse to the folder bin and type ./mstrtodbcx, then click Enter. Prompts guide you through testing your ODBC connection from the command line and should be used in
place of the steps below. For detailed steps on how to use the command line version of this tool, see *Testing ODBC connectivity* in Chapter 12, *Configuring MicroStrategy Using Command Line Tools*.

2. From the **Session** menu, select **Open Connection**, or click the **Connect** icon on the toolbar. The Connect dialog box opens. The connection interface varies depending on the destination database.

3. Select the DSN for a data source.

4. Enter the appropriate user name and password.

5. Click **Connect**. After your connection is opened, the connection string is displayed in the MicroStrategy DB Query Tool at the bottom. Your cursor is inserted automatically in the SQL Statement window.

6. In the SQL Statement window, type a SQL query such as:

   ```sql
   select count (*) from Table

   where Table is a system-defined table, such as SYSOBJECTS for Microsoft SQL Server or a MicroStrategy-created table such as DSSMDSYSPROP in the MicroStrategy metadata.
   ```

7. From the **Queries** menu, select **Execute Query**. A table of data from the database is displayed in the Query Result window.

8. From the **Session** menu, select **Close Connection** to close the database connection.

9. From the **File** menu, select **Exit**.
Initial MicroStrategy configuration

The MicroStrategy Configuration Wizard automates much of the configuration process, prompting you only when information is required. With this tool, you can configure the metadata repository, statistics tables and Enterprise Manager repository, History List tables, MicroStrategy Intelligence Server, and multiple project sources.

If you are configuring MicroStrategy using the Windows operating system, you must have administrative privileges on the computer on which the Intelligence Server is installed, so that you can define the parameters necessary to start the Intelligence Server and to invoke server-definition objects.

You can also configure your MicroStrategy installation using the Configuration Wizard in silent or unattended mode. This allows you to load an existing setup routine to configure your MicroStrategy installation. For information on running the Configuration Wizard with a response file, see Configuring MicroStrategy with a Response File, page 404.

You can also use the Configuration Wizard to create an Enterprise Manager project, which provides insights about governing and tuning all areas of your MicroStrategy environment. For steps on how to create an Enterprise Manager project, see the Enterprise Manager Guide.

Configuration Wizard Prerequisites

Before you begin using the Configuration Wizard you should review and complete the following requirements:

- Install the necessary MicroStrategy products. You should have at least MicroStrategy Developer and MicroStrategy Intelligence
Server installed. For information on how to install MicroStrategy on Windows, see *Chapter 2, Installing MicroStrategy on Windows*. For information on how to install MicroStrategy on other operating systems, see *Chapter 3, Installing MicroStrategy on Linux*.

- Have access to an empty database location certified to house the metadata. This includes creating DSNs for your databases (see *Communicating with Databases, page 356*). For a list of certified metadata platforms, see the *MicroStrategy Readme*.

- In a Linux environment, the Configuration Wizard must be able to communicate with Intelligence Server over TCP/IP network protocol. To achieve this, the `hosts` file in the `/etc` directory must include one entry identifying the local host in the form:

  \[ IP-address \text{ local-machine-name} \]

  For example, `123.4.5.6 FakeISmachine`. Modifying the `hosts` file may require an account with root privileges.

- MicroStrategy products must be configured on the machine on which they are installed. You cannot configure them remotely.

**Configuring MicroStrategy Software**

The MicroStrategy Configuration Wizard opens automatically after you install MicroStrategy products and restart your machine.

You can configure a MicroStrategy Web and Intelligence Server (four-tier), Intelligence Server (three-tier), or direct (two-tier) setup for MicroStrategy.

The following figure describes how to configure MicroStrategy to suit an Intelligence Server (three-tier) environment. It also shows how the various components of the MicroStrategy Configuration Wizard, the metadata repository, Intelligence Server, and the project source interact with each other.
A MicroStrategy Web (four-tier) setup involves configuring a web server to communicate between Intelligence Server and MicroStrategy Web. For more information on deploying MicroStrategy Web, see Chapter 7, Deploying MicroStrategy Web and Mobile Server.

The figure below describes how to configure MicroStrategy to suit a direct (two-tier) environment. It also shows how the various components of the MicroStrategy Configuration Wizard, the metadata repository and the project source interact with each other:
It is not recommended to use a direct setup for the production environment.

The procedure below provides the high-level steps to configure MicroStrategy software through the Configuration Wizard.

To configure MicroStrategy through the Configuration Wizard

If you are configuring MicroStrategy on:

- Windows, go to Start > Programs > MicroStrategy Tools > Configuration Wizard. Continue to the steps provided in To select a configuration task, page 374.

- Windows from the command line, then perform the following steps:
  
  - From the Windows Start menu, select Run.
  
  - In the Open drop-down list, type cmd and click OK. A command prompt opens.
  
  - Type macfgwiz and click Enter.

  This command displays the command line version of the Configuration Wizard. You can configure the connection of a data source to Intelligence Server by creating a response file or using an existing response file. The command line prompts guide you through configuring the connection of a data source to Intelligence Server by creating a response file or using an existing response file and should be used in place of the steps below. For information on using a response file to configure MicroStrategy, see Configuring MicroStrategy with a Response File, page 404.

- Linux using the Configuration Wizard interface, then perform the following steps:
From a 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 ./mstrcfgwiz, then click Enter. The Configuration Wizard opens.

Linux from the command line, then perform the following steps:

- From a 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 ./mstrcfgwiz-editor, then click Enter.

This command displays the command line version of the Configuration Wizard. You can configure the connection of a data source to Intelligence Server by creating a response file or using an existing response file. The command line prompts guide you through configuring the connection of a data source to Intelligence Server by creating a response file or using an existing response file and should be used in place of the steps below. For information on using a response file to configure MicroStrategy, see Configuring MicroStrategy in Command Line Mode, page 679.

To select a configuration task

Choose from the following configuration tasks and then click Next to begin the selected task:

- **Create Metadata, History List and Enterprise Manager Repositories**: Runs the SQL scripts necessary to create and initialize the metadata repository, History List tables, and Enterprise Manager statistics tables and repositories in the database location
that you select. For steps to complete these configuration tasks, see *Creating Metadata, History List, and Statistics Repositories, page 376.*

- **Configure Intelligence Server:** Creates a new server definition object in the metadata repository that you select. This setup provides Intelligence Server (three-tier) access to all projects that are stored in the repository. This option also allows you to use or delete an existing server definition. For steps to complete these configuration tasks, see *Setting up MicroStrategy Intelligence Server, page 389.*

- **Create Enterprise Manager project:** The Enterprise Manager project provides insights about governing and tuning all areas of your MicroStrategy environment. For steps on how to create an Enterprise Manager project, see the *Enterprise Manager Guide.*

- **Create Platform Analytics project:** The Platform Analytics project provides insights about governing and tuning all areas of your MicroStrategy environment.

- **Create a Project Source:** A project source contains the configuration information that each client system requires to access an existing project. It stores the location of the metadata repository and Intelligence Server that is used to run the project. A project source determines how MicroStrategy Developer, Web, and other client applications access the metadata. For steps to complete these configuration tasks, see *Creating Project Sources, page 398.*

- **Upgrade existing environment to MicroStrategy Intelligent Enterprise:** You can use the Configuration Wizard to upgrade your MicroStrategy environment and migrate various features to the new version. For all available upgrade and migration options, see the *Upgrade Guide.*

The remainder of this section describes each configuration option in detail.
After completing these steps, an empty metadata repository is created. To learn how to add projects to your metadata repository, see the Project Design Guide.

Creating Metadata, History List, and Statistics Repositories

You can create metadata, History List, and statistics and Enterprise Manager repositories using the MicroStrategy Configuration Wizard. Repositories for your metadata, History List, and statistics tables are created in the data source specified by the DSN(s) you connect to.

It is recommended that you create the metadata, History List, and statistics repository tables in different databases to ensure enhanced performance.

As you complete the configuration process, messages may be displayed. For details on system messages displayed during the configuration process, see Configuration messages, page 389.

You can choose to create metadata, History List, and statistics repositories using a response file with the Configuration Wizard. This lets you provide users with a configuration file to complete the configuration tasks rather than requiring users to step through the Configuration Wizard. Creating and using a response file can be done using the Configuration Wizard interface or a command line tool available for Linux.

Required database permissions to create metadata, History List, and statistics repositories

To create metadata, History List, and statistics repositories in a database, you need a database user account to associate with the tables created for the repositories. MicroStrategy recommends that
the database user account used to create these repositories is granted full permissions for the database.

If the database user account cannot be granted full permissions to the database, be aware that this account requires Select, Insert, Update, Create, Drop, and Delete permissions. These permissions are required for various database objects depending on the database type you are using. For example, the following database object permissions are required where applicable for your database type:

<table>
<thead>
<tr>
<th>Database Object</th>
<th>Type of Permissions Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>Select, Insert, Update, Create, Drop, Delete</td>
</tr>
<tr>
<td>Indexes</td>
<td>Create, Drop</td>
</tr>
<tr>
<td>Triggers</td>
<td>Create, Drop</td>
</tr>
<tr>
<td>Functions</td>
<td>Create, Execute</td>
</tr>
<tr>
<td>Packages</td>
<td>Create</td>
</tr>
<tr>
<td>Procedures</td>
<td>Create, Execute</td>
</tr>
</tbody>
</table>

While creating metadata, History List, and statistics repositories, the Configuration Wizard provides an option to preview the SQL statements that will be executed. You can review this SQL preview to have a better understanding of the tasks that will be required as part of creating metadata, History List, and statistics repositories.

Refer to your third-party database documentation for specific names and details on database permissions and database objects.

Creating a metadata repository

The metadata repository is a collection of tables that contain the definitions for nearly all MicroStrategy objects including database logins, server definitions, database instances and connections,
reports, metrics, facts, and so on. It is mandatory to have a metadata repository to which Intelligence Server can connect.

You can create the metadata repository in the database location of your choice. Additionally, a default configuration is created in the metadata tables. This populates the tables with the basic data required for the MicroStrategy metadata, such as the default project folder structure and some basic connection information.

If the Intelligence Server is not installed on the same machine as Configuration Wizard, you can only create a metadata repository using the 32-bit Configuration Wizard.

If you are upgrading your metadata from a previous version of MicroStrategy rather than creating a brand new metadata, see the Upgrade Guide.

Before you create a metadata repository, you should ensure that you are storing it on a certified database, ODBC driver, and operating system combination. For a list of certified metadata repository environments, see the MicroStrategy Readme.

- A database user account to associate with the metadata tables. MicroStrategy recommends that the database user account used to create a metadata repository is granted full permissions for the database. If the database user account cannot be granted full permissions to the database, refer to Required database permissions to create metadata, History List, and statistics repositories, page 376 for additional details on the database permissions required for this configuration.

- While metadata creation errors are rare in general, you can review a list of potential errors in Metadata and Other Repository Creation Errors, page 917 to prepare for or avoid specific scenarios that can cause errors.
To create a metadata repository

1. Open the MicroStrategy Configuration Wizard.

2. Select **Metadata, History List and Statistics Repository Tables** and click **Next**.

3. Select the **Metadata Tables** check box and click **Next**. The Repository Configuration: Metadata tables page opens.

   You can also select to create a History List and a statistics repository immediately after creating a metadata repository. If you create a History List or statistics repository as part of the same configuration routine as creating a new metadata repository, and the configuration is being done on a Windows environment, database instances are automatically created for the History List and statistics repositories.

4. From the **DSN** drop-down list, select the DSN for your metadata repository.

   If a DSN for your metadata repository does not exist, you can click **New** to open the Connectivity Wizard and create a new DSN. The steps to create a new DSN with the Connectivity Wizard are described in *Creating a DSN for a data source, page 363*.

   Although it is possible to use the Microsoft Access database for the metadata repository, it is not a suitable metadata repository for a production project. You should not use Microsoft Access for anything other than a proof-of-concept or demonstration type of application.

5. Type a **User Name** and **Password** that can connect to the data source.
The database user that you provide becomes the owner of all metadata tables and objects. The database user is required to have the Select, Insert, and Update permissions. Intermediate tables are created in the metadata for recursive search queries, which requires Create and Drop permissions as well. Updating the schema requires the Delete permission.

6. After providing a valid user name and password, you can click **SQL Preview** to open the SQL Preview dialog box. This dialog box provides the SQL statements that will be executed on your data source to create the metadata tables. Click **Close**.

   If you use the advanced options to change the SQL script, you can click SQL Preview after selecting the new script to see an updated listing of the SQL statements that will be executed.

7. Click **Advanced**.

8. In the **Table Prefix** field, you can specify a prefix to be used when metadata tables are created in the database you select. This is an optional configuration. However, you must use different prefixes for your metadata tables and your History List tables if you store them in the same database.

   Most databases use a prefix of two characters. However, you can supply as many letters, numbers, underscores (_), and periods (.) as required to support your database prefixes. To determine character limits for a prefix, refer to your third-party database vendor documentation.

9. In the **Script** field, a SQL script to create metadata tables optimized for your database is selected. If you want to select a different script, click ... to browse to and select a customized script.
10. Click **Next**. The next configuration page that opens depends on your configuration scenario:

- If your metadata repository does not need to be upgraded and you did not select to create History List or statistics tables, the Summary page opens. You can complete your configuration as described in *Creating Metadata, History List, and Statistics Repositories, page 376* below.

- If your metadata repository does not need to be upgraded and you selected to configure History List or statistics tables, you are prompted to configure these options as described in *Creating a History List repository, page 381* and *Creating statistics and Enterprise Manager repositories to maintain and monitor system activity, page 385*.

- If your metadata repository needs to be upgraded, cancel this metadata creation process. If you continue with this process of creating metadata tables, your current metadata will be overwritten with a brand new metadata. For information on upgrading your metadata and suite of MicroStrategy projects, refer to the **Upgrade Guide**.

11. Review the summary information.

   You can click **Save** to save the configuration as a response (.ini) file to configure metadata repositories on other systems or to run silent configurations at a later time.

12. Click **Finish**.

**Creating a History List repository**

A History List repository stores users' report and document results for future use. History Lists can be stored on file systems of a server machine. The History List tables provide an alternative option to store History List information in a centralized database.
If you create a History List repository as part of the same configuration routine to create a metadata repository, and the configuration is being done on a Windows environment, a database instance is automatically created for the History List repository. If you create the History List repository separately, you create it for an existing metadata repository, or you create it on a Linux environment, you must create a database instance for the History List repository. For information on creating a database instance, see *Creating a database instance, page 435.*

If you are upgrading your History List repository from a previous version of MicroStrategy rather than creating a brand new History List repository, see the Upgrade Guide.

Before you create a History List repository, you should ensure that you are storing it on a certified database, ODBC driver, and operating system combination. For a list of certified History List repository environments, see the *MicroStrategy Readme.*

- A database user account to associate with the History List tables. MicroStrategy recommends that the database user account used to create History List Tables is granted full permissions for the database. If the database user account cannot be granted full permissions to the database, refer to *Required database permissions to create metadata, History List, and statistics repositories, page 376* for additional details on the database permissions required for this configuration.

- The steps below are specific to creating a History List repository. If you also select to create a metadata repository, you must first complete the steps described in *Creating a metadata repository, page 377.*

- While History List creation errors are rare in general, you can review a list of potential errors in *Metadata and Other Repository Creation Errors, page 917* to prepare for or avoid specific scenarios that may cause errors.
To create a History List repository

1. Open the MicroStrategy Configuration Wizard.

2. Select **Metadata, History List and Statistics Repository Tables** and click **Next**.

3. Select the **History List Tables** check box and click **Next**.

4. From the **DSN** drop-down list, select the DSN for your History List repository.

   If a DSN for your History List repository does not exist, you can select **New** to open the Connectivity Wizard and create a new DSN. The steps to create a new DSN with the Connectivity Wizard are described in *Creating a DSN for a data source, page 363*.

5. Type a **User Name** and **Password** that can connect to the data source.

   The database user you provide becomes the owner of all History List tables and objects. The database user is required to have the Select, Create, Insert, and Drop permissions.

6. After providing a valid user name and password, you can click **SQL Preview** to open the SQL Preview dialog box. This dialog box provides the SQL statements that will be executed on your data source to create the History List tables. Click **Close** once you are done reviewing the SQL statements to return to the Configuration Wizard.

   If you use the advanced options to change the SQL script, you can click SQL Preview after selecting the new script to see an updated listing of the SQL statements that will be executed.

7. Click **Advanced**.
8. In the **Table Prefix** field, you can specify a prefix to be used when History List tables are created in the database you select. This is an optional configuration. However, you must use different prefixes for your metadata tables and your History List tables if you store them in the same database.

Most databases use a prefix of two characters. However, you can supply as many letters, numbers, underscores (_), and periods (.) as required to support your database prefixes. To determine character limits for a prefix, refer to your third-party database vendor documentation.

If you use a table prefix for your History List tables, you must also define this table prefix when you create a database instance to connect to the History List tables. For information on creating a database instance, see *Creating a database instance, page 435*.

9. In the **Script** field, a SQL script to create History List tables optimized for your database is selected. If you want to specify a different script, click ... to browse to and select a customized script. For more information on the default SQL scripts, see *SQL scripts, page 389*.

10. Click **Next**. The next configuration page that opens depends on your configuration scenario:

- If you did not select to create statistics tables, the Summary page opens.

- If you selected to configure statistics tables, you are prompted to configure these options as described in *Creating statistics and Enterprise Manager repositories to maintain and monitor system activity, page 385*. 
11. Review the summary information.

You can click **Save** to save the configuration as a response (.ini) file to configure History List repositories on other systems or to run silent configurations at a later time. For information on running the Configuration Wizard with a response file, see *Configuring MicroStrategy with a Response File, page 404.*

12. Click **Finish** to apply the configuration and create the History List repository.

Once you are finished configuring Intelligence Server and your project sources, a database instance to connect a project to a History List repository must be created. If you created the History List repository as part of the same configuration routine to create a metadata repository and the configuration is being done on a Windows environment, a database instance is automatically created for the History List repository. For information on creating a database instance, see *Creating a database instance, page 435.*

Creating statistics and Enterprise Manager repositories to maintain and monitor system activity

The statistics and Enterprise Manager repositories are collections of database tables used to maintain and monitor system activity and performance. You can run MicroStrategy Enterprise Manager against the statistical information to analyze and interpret the statistics.

For a detailed description of the statistics tables used in the Enterprise Manager Statistics database, the fields that each table contains, and the data types associated with each field for MicroStrategy Intelligence Server, refer to the *Enterprise Manager Data Model and Object Definitions* chapter in the *System Administration Guide.*
For details on how to configure projects to log statistics, refer to the *Monitoring System Usage* chapter in the *System Administration Guide*.

If you create statistics and Enterprise Manager repositories as part of the same configuration routine to create a metadata repository, and the configuration is being done on a Windows environment, a database instance is automatically created for the statistics repository. If you create the statistics repository separately, you create it for an existing metadata repository, or you create it on a Linux environment, you must create a database instance for the statistics repository. For information on creating a database instance, see *Creating a database instance, page 435*.

If you are upgrading your statistics and Enterprise Manager repositories from a previous version of MicroStrategy rather than creating a brand new statistics repository, see the *Upgrade Guide*.

Before you create statistics and Enterprise Manager repositories, you should ensure that you are storing them on a certified database, ODBC driver, and operating system combination. For a list of certified environments, see the *MicroStrategy Readme*.

- A database user account to associate with the repositories. MicroStrategy recommends that the database user account used to create the tables is granted full permissions for the database. If the database user account cannot be granted full permissions to the database, refer to *Required database permissions to create metadata, History List, and statistics repositories, page 376* for additional details on the database permissions required for this configuration.

- The steps below are specific to creating statistics and Enterprise Manager repositories. If you also select to create a metadata repository or History List repository, you must first complete the steps described in
Creating a metadata repository, page 377 or Creating a History List repository, page 381, respectively.

- While statistics creation errors are rare in general, you can review a list of potential errors in Metadata and Other Repository Creation Errors, page 917 to prepare for or avoid specific scenarios that may cause errors.

To create statistics and Enterprise Manager repositories

1. Open the MicroStrategy Configuration Wizard.

2. Select Metadata, History List and Statistics Repository Tables and click Next.

3. Select the Statistics & Enterprise Manager Repository check box and click Next.

4. From the DSN drop-down list, select the DSN for your statistics and Enterprise Manager repositories.
   
   If an applicable DSN does not exist, you can select New to open the Connectivity Wizard and create a new DSN. The steps to create a new DSN with the Connectivity Wizard are described in the section Creating a DSN for a data source, page 363.

5. Type a User Name and Password that can connect to the data source.

   The database user you provide becomes the owner of all tables and objects. The database user is required to have the Select, Create, Insert, and Drop permissions.

6. After providing a valid user name and password, you can click SQL Preview to open the SQL Preview dialog box. This dialog box provides the SQL statements that will be executed on your data source to create the statistics and Enterprise Manager
tables. Click Close.

If you use the advanced options to change the SQL script, you can click SQL Preview after selecting the new script to see an updated listing of the SQL statements that will be executed.

7. Click Advanced.

8. In the Script field, a SQL script to create statistics and Enterprise Manager repositories optimized for your database is selected. If you want to specify a different script, click ... (the browse button) to browse to and select a customized script. For more information on the default SQL scripts, see SQL scripts, page 389.

9. Click Next.

10. Review the summary information.

You can click Save to save the configuration as a response (.ini) file to configure statistics repositories on other systems or to run silent configurations at a later time. For information on running the Configuration Wizard with a response file, see Configuring MicroStrategy with a Response File, page 404.

11. Click Finish. The summary information is updated as the configurations are completed, providing a way to track the progress of the configurations.

Once you are finished configuring Intelligence Server and your project sources, a database instance to connect a project to a statistics repository must be created. If you created the statistics repository as part of the same configuration routine to create a metadata repository and the configuration is being done on a Windows environment, a database instance is automatically created for the statistics repository. For information on creating a database instance, see Creating a database instance, page 435.
SQL scripts

MicroStrategy has database-specific SQL scripts for creating metadata, History List, and statistics tables. The scripts for each certified database platform are shipped with the product. The MicroStrategy Configuration Wizard automatically selects a default script based on your ODBC driver's database platform.

By default, all the scripts reside in the directory where you installed MicroStrategy and are identified by the .sql extension. It is highly recommended that no edits be performed on these scripts, except on rare occasions and only by skilled database personnel.

Configuration messages

Depending on the selected ODBC database, different messages might be displayed prompting you to complete the configuration successfully. Two examples are described below:

- Metadata tables already exist at this location. Would you like to recreate them? (This will drop all existing information in the Metadata)

  This message is displayed if the Configuration Wizard detects an existing metadata repository in the database location you specified.

  ! If you continue, all information in the existing metadata repository is overwritten.

- No Metadata Tables were found at this location, do you wish to create them now?

  This message is displayed if there is no existing metadata repository and you have not chosen to create one.

Setting up MicroStrategy Intelligence Server

You use the Configuration Wizard to create and configure a server definition for your MicroStrategy Intelligence Server. A server
definition is stored in the metadata repository, and it contains information about the configuration of Intelligence Server such as governing settings, which projects should be loaded, which communication protocols should be used, and so on. This definition is a required step of configuring your Intelligence Server.

Multiple server definitions can be available, but you can install only one Intelligence Server on one server machine and Intelligence Server uses only one server definition at a time.

You can choose to configure the server definition, project source names, and the metadata and statistics repositories using a response file with the Configuration Wizard. This lets you provide users with a configuration file to complete the configuration tasks rather than requiring users to step through the Configuration Wizard. Creating and using a response file can be done using the Configuration Wizard interface or a command line tool available for Linux.

You must run the Configuration Wizard locally on the Intelligence Server machine. You cannot create, use, or delete server definitions remotely.

To set up MicroStrategy Intelligence Server

1. Open the MicroStrategy Configuration Wizard.
2. Select Configure Intelligence Server, and click Next.
3. From the DSN drop-down list, select the DSN used to connect to the metadata repository. Enter the User Name and Password for the database. Although it is possible to use the Microsoft Access database for the metadata repository, it is not a suitable metadata repository for a production project. You should not use Microsoft Access for anything other than a proof-of-concept or demonstration type of application.
4. Click **Next**. If a message is displayed that your metadata is not up to date with the most recent version of MicroStrategy, you must upgrade your metadata to take advantage of the new features available in the most recent version of MicroStrategy. You can upgrade your MicroStrategy metadata as described in the Upgrade Guide.

5. On the MicroStrategy Authentication page, specify the MicroStrategy administrator's **User Name** and **Password**. By default, the user name is Administrator and it has no password. If you are setting up Intelligence Server for the first time, use the default user name and password.

   For security reasons, you should change the Administrator user name and password as soon as possible after you initially configure the system. For details about passwords and other user management information, see the System Administration Guide.

6. Click **Next**.

7. You can create a new server definition, use an existing server definition, or delete a current server definition. You can perform one of the following tasks:

   - To create a server definition, select **Create New Server Definition**. When you create a new server definition in the metadata repository of your choice, all its parameters use the default settings. You can modify these default settings using the MicroStrategy Intelligence Server Configuration Editor. For information about the Intelligence Server Configuration Editor, see the System Administration Guide.

     1. In the **Name** field, type a name to identify the server definition.
2. Select the **Use as the active server definition** check box to define Intelligence Server to use the new server definition when Intelligence Server starts.

3. Click **Next**. The Server Configuration: Settings page opens, described in *Setting up MicroStrategy Intelligence Server, page 389* below.

- To use an existing server definition, select **Use the selected Server Definition as active**. When you use an existing server definition different from the current server definition, you are changing the machine's configuration information, which can be in an entirely different metadata with different default settings.

  1. From the Existing Server Definitions pane, select a server definition to use.

  2. Click **Next**. The Server Configuration: Settings page opens, described in *Setting up MicroStrategy Intelligence Server, page 389* below.

- To delete an existing server definition, select **Delete Selected Server Definition**. When you delete a server definition, you are deleting the server definition object from the metadata repository, but not from the Intelligence Server software that you installed.

  1. From the Existing Server Definitions pane, select a server definition to delete.

  2. Click **Next**. The Summary page opens, described in *Setting up MicroStrategy Intelligence Server, page 389* below.

Select the server definition to use from the **Existing Server Definitions** pane. Click **Next**.
Define the Intelligence Server port number and other settings

1. If you select to create a server definition or use an existing server definition, you can define the Intelligence Server port number and other settings, as described below:

   - **Port number**: You can use the default port number (34952) or specify another port number. The port number is how a server process identifies itself on the machine on which it is running. If the port number is used by another process, such as in a shared environment, specify an available port number. For instructions on how to find an available port number, see *Port Number is in Use, page 916*.

   - **REST port number**: You can use the default port number (34962) or specify another port number for the REST API Server inside Intelligence Server. This port number should not be same with the Intelligence Server port number.

      You can configure REST API Server logging with the Diagnostics and Performance Logging Tool. Select the **Performance Configuration** tab, and find the dispatcher **REST Trace** under component **Network Classes**. For more information, see the Configuring What is Logged section in the *System Administration Guide*.

   - **Register Intelligence Server as a Service**: This option is only available if you are configuring Intelligence Server on a Linux machine, and you have root access and permissions to the Linux machine that Intelligence Server is installed on. In Windows, Intelligence Server is automatically registered as a service upon completion of the configuration process. Running the Configuration Wizard again and clearing this check box does not unregister Intelligence Server as a service. To unregister Intelligence Server on Linux, you must stop the
service, and then use the mstrctl command line tool. The syntax is mstrctl -s IntelligenceServer us, where IntelligenceServer is the name of a server definition. For information about starting, stopping, and registering Intelligence Server as a service, see the System Administration Guide.

- **Projects to load at startup**: This pane displays all the projects that are in the metadata repository. You can select projects to use with the server definition that you have chosen. The projects that you select are loaded on the server at startup.

- **Start Intelligence Server when finished**: Select this check box to have Intelligence Server start once you complete this configuration. If you use Windows NT authentication with SQL Server, you must type the Windows NT account user name and password in the Service Manager to successfully start Intelligence Server. For information on how to access and use the Service Manager, see the System Administration Guide.

- **Identify missing DSNs**: Select this check box to verify that all DSNs, which are used for database instances created in MicroStrategy, are locally available. This helps to ensure that your database instances in MicroStrategy can connect successfully to their associated data sources.

  By default, this check box is cleared, which means the availability of all local DSNs used in database instances is not verified. While this may mean that all DSNs used in database instances are not available, it can save system resources required for the Intelligence Server configuration process.

2. Click **Next**.
3. You can enable or disable secure socket layer (SSL) protocol to encrypt the communication between Intelligence Server and Developer:

**Configure SSL:** This option specifies whether to enable Intelligence Server and Developer to communicate using the SSL protocol. Clear this check box to disable the use of the SSL protocol for Intelligence Server and Developer communications. This option also applies SSL protocol to the REST port number.

When you select to enable the SSL protocol, you must provide the following information:

- **Certificate:** The SSL certificate file you created for Intelligence Server. Click the browse button to navigate to and select the certificate file.

- **Key:** The private key file you created while requesting the certificate for Intelligence Server. Click the browse button to navigate to and select the private key file.

- **Password:** The password that you used while creating the private key for the SSL certificate.

- **SSL Port:** The port number to use for SSL access. By default, the port is 39321.

To enable SSL protocol communication in Developer, you must use the Project Source Editor. For steps to complete the other tasks required to enable SSL protocol communications, refer to the System Administration Guide.

**Configuring port requires Client Certificate:** If selected, the SSL communications with client certificate verification will be
configured in the Intelligence Server, but does not apply to REST port number. Provide the following information:

- **SSL Port**: The port number the Intelligence Server will use for SSL communications with client certificate verification.

- **Truststore**: The location to the client certificate truststore.

4. Click **Next**.

5. You can specify the default statistics repository to use for the local Intelligence Server, including the data source name, user name, and password, and an option to create a new data source name. The following options appear:

- **Keep the existing statistics settings**: Select this check box to keep the existing statistics settings for all projects. This option is selected by default.

- **Enable Statistics Database Instance for the Intelligence Server metadata**: Select this check box to enable basic statistics logging for projects.

   You can alter the settings of this option in the Project Configuration Editor in MicroStrategy Developer. For steps, see the System Administration Guide.

When defining the default statistics repository, you must provide the following configuration details:

- **DSN**: Select the data source name for your statistics repository.

   If a DSN for your statistics repository does not exist, you can click **New** to open the MicroStrategy Connectivity Wizard and create a new DSN. Unsupported DSNs are grayed out.
- **User Name**: Enter the database user name for the user that can connect to the statistics data source.

- **Password**: Enter the password for the user that can connect to the statistics data source.

- **Disable statistics for the local Intelligence Server metadata**: Select this check box to disable statistics settings for all projects in the local Intelligence Server metadata.

6. Click **Next**.

7. You can either enable or disable Messaging Services for Intelligence Server and configure the Messaging Services host and port settings.

   **Messaging Services Host(s)/Port(s)**: Provide host name or IP address of Messaging Services. Format for this setting should be: `server1.acme.com:9092, server2.acme.com:9093, server3.acme.com:9093`.

8. Click **Next**.

9. Review the summary information and click **Save**.

   For information on running the Configuration Wizard with a response file, see *Configuring MicroStrategy with a Response File, page 404*.

10. Click **Finish**.

    If you created a new server definition, it is displayed in the list of existing server definitions for that metadata.

    If you assigned an existing server definition to Intelligence Server and the existing project source uses this Intelligence Server, a related message is displayed.
Starting, stopping, and restarting the server

With a server definition defined for your Intelligence Server, you can use Service Manager to start or stop your Intelligence Server. For steps to use Service Manager, see the System Administration Guide.

Creating Project Sources

Project sources represent a connection to a metadata database or a MicroStrategy Intelligence Server. The project source stores the location of the metadata repository or the MicroStrategy Intelligence Server definition that is used to run the project. Through a project source you can create, manipulate, and administer MicroStrategy projects.

When you create a metadata repository, by default it creates a server (three-tier) project source. You can use the Project Sources option in the MicroStrategy Configuration Wizard if you need to create multiple project sources or a direct (two-tier) project source. The steps to create the different types of project sources are:

- **Creating a direct (two-tier) project source, page 399**: Direct project sources that connect directly to the metadata through ODBC. You cannot create a direct project source on Linux.

- **Creating a server (three-tier) project source, page 400**: Server project sources that connect to the metadata through an Intelligence Server.

You can choose to create project sources using a response file with the Configuration Wizard. This lets you provide users with a configuration file to complete the configuration tasks rather than requiring users to step through the Configuration Wizard. Creating and using a response file can be done using the Configuration Wizard interface or a command line tool available for Linux.
Creating a direct (two-tier) project source

A direct project source is used to connect directly to the metadata repository using ODBC. A direct project source connection does not allow you to access MicroStrategy Web, run Report Services documents, or use any of the other MicroStrategy features that are provided through Intelligence Server.

You cannot create a direct project source on Linux.

For Windows, the Project Source option is available only if the Developer product is installed on the machine.

To create to a direct project source

1. Open the MicroStrategy Configuration Wizard.

2. Select Project Sources and click Next.

3. In the Project Source Name field, type a name for the project source.

4. Go to Connection Type > Direct (2-tier), and click Next.

5. From the DSN drop-down list, select a DSN for the data source that stores the metadata and specify a User Name and Password.

6. You can also click New to create a new DSN (see Creating a DSN for a data source, page 363) and click Advanced to specify a metadata table prefix if necessary.

7. Click Next.

8. Select the authentication mode for the project source. For information on the available authentication modes, see the Authentication modes, page 402.

9. Click Next.
10. Review the summary information.

You can click **Save** to save the configuration as a response (.ini) file to configure a direct project source on other systems or to run silent configurations at a later time. For information on running the Configuration Wizard with a response file, see *Configuring MicroStrategy with a Response File, page 404*.

11. Click **Finish**.

Creating a server (three-tier) project source

A server (three-tier) project source is used to connect to the metadata using the MicroStrategy Intelligence Server. A server project source connection allows you to access MicroStrategy Web, run Report Services documents, and use all of the other MicroStrategy features that are provided through Intelligence Server.

When you create a metadata repository, by default it creates a server (three-tier) project source.

For Windows, the Project Source option is available only if the Developer product is installed on the machine.

To create a MicroStrategy Intelligence Server (three-tier) project source

1. Open the MicroStrategy Configuration Wizard.
2. Select **Project Sources** and click **Next**.
3. In the **Project Source Name** field, type a name for the project source.
4. Under **Connection Type**, select **MicroStrategy Intelligence Server (3-tier)**, and click **Next**.
5. In the **MicroStrategy Intelligence Server Machine Name** drop-down list, select the Intelligence Server to connect to.

6. In the **Port Number used by MicroStrategy Intelligence Server** field, type the port number for the Intelligence Server to connect to.

   The port number is how the Intelligence Server process identifies itself on the server on which it is running. The default port number for Intelligence Server is 34952. If you use a non-default port number, this number must be provided while connecting through MicroStrategy Developer.

   If you set up a firewall between Intelligence Server and your MicroStrategy Web server, refer to the **System Administration Guide** for steps to ensure the required ports are open to allow communication between your MicroStrategy systems.

7. Select the **Connection times out after (mins) check box** to define and enforce a connection time out for inactive users connected to a project source. In the field below, type a numerical value (in minutes) for the amount of inactivity that is allowed before a user is automatically disconnected from a project source. If this check box is cleared, users are not disconnected from project sources due to inactivity.

8. Click **Next**.

9. Select the authentication mode for the project source. For information on the available authentication modes, see the **Authentication modes, page 402**.

10. Click **Next**.
11. Review the summary information.

You can click **Save** to save the configuration as a response (.ini) file to configure a server project source on other systems or to run silent configurations at a later time. For information on running the Configuration Wizard with a response file, see *Configuring MicroStrategy with a Response File, page 404*.

12. Click **Finish**.

Authentication modes

Authentication is the process through which the system identifies the user. Several authentication modes are supported for MicroStrategy project sources. They vary primarily by the system that verifies and accepts the login/password credentials provided by the user.

Some authentication modes require a server project source (three-tier). Therefore, if you are creating a direct project source (two-tier) some of the authentication options listed below cannot be used.

For information on the benefits of the various authentication modes and other authentication topics, see the *System Administration Guide*

Network login ID: Windows authentication

To use Windows authentication, you must create users in the MicroStrategy environment and then link them to Windows users. If you use Windows as your network operation system and your users are already defined in the Windows directory, your users can access the MicroStrategy application without having to enter a login ID and password.
Login ID and password entered by the user: Standard authentication

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. Each user has a unique login/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 information on configuring connection mapping, see the System Administration Guide.

Guest account: Anonymous authentication

When using anonymous authentication, users log in as Guest and do not need to provide a password. By default, guest users can access the project, browse objects, run and manipulate reports, but they cannot create their own objects or schedule report executions. However, you determine what the Guest user can and cannot do by modifying the Public user group. Guest users inherit their privileges from the Public group; they are not part of the Everyone group.

LDAP authentication

Lightweight Directory Access Protocol (LDAP) authentication identifies users within a repository of users stored in an LDAP server (such as Novell Directory Services). If you use an LDAP directory to centrally manage users in your environment, you may want to use LDAP authentication. Group membership can be maintained in the LDAP directory without having to also be defined in the MicroStrategy Intelligence Server. When using LDAP authentication, LDAP users or
groups are linked to users or groups in the MicroStrategy environment.

Login ID and password entered by the user for the warehouse: Database authentication

This mode of database authentication identifies users using a login ID and password stored in the data warehouse database. Under this mode of authentication, a warehouse database is associated with each project. When users log in to a project source, they are logging in to the Intelligence Server. Use database 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 the Intelligence Server as well as the RDBMS.

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

Configuring MicroStrategy with a Response File

The Configuration Wizard walks you through the process of setting up the environment for the MicroStrategy products installed in your system. You can also configure server definitions, project source names, an Enterprise Manager project, and the metadata, History List, and statistics repositories using a response file with the Configuration Wizard. This enables you to provide a configuration file to users to complete the configuration tasks, rather than requiring users to step through the Configuration Wizard. This can be done to configure a MicroStrategy installation on Windows and Linux.
The Configuration Wizard can also be used to perform MicroStrategy upgrades and create an Enterprise Manager project. These tasks can also be accomplished by using a response file:

- For steps to upgrade MicroStrategy using a response file, see the Upgrade Guide.
- For steps to create an Enterprise Manager project with a response file, see the Enterprise Manager Guide.

Creating a response file

It is recommended that you always create the response file through the graphical interface of the Configuration Wizard, as described in the procedure To create a response file, page 405 in this section. This ensures that all applicable options are included in the response file with valid values.

However, you can also create and use a response file with the Configuration Wizard in command line mode on Linux machines. For steps to create and use a response file as well as perform other configurations using command line tools in Linux, see the Configuring MicroStrategy with a response.ini file, page 703 section in Chapter 12, Configuring MicroStrategy Using Command Line Tools.

To create a response file

1. Open the MicroStrategy Configuration Wizard. To do this, see To configure MicroStrategy through the Configuration Wizard, page 373.

2. Any configuration tasks you complete with the Configuration Wizard can be saved to a response file. For steps to complete various configurations tasks with the Configuration Wizard, see the sections listed below:
Steps to use a response file to configure MicroStrategy are covered in 
Using a response file to configure MicroStrategy installations, page 406 below.

You can modify a response file with a text editor to make
configuration changes such as entering different user login and
password information. For information on the parameters and options
available in response files, see Response configuration parameters and options, page 408.

Using a response file to configure MicroStrategy installations

Rather than stepping through each page of the Configuration Wizard,
you can configure MicroStrategy using a response file. You have the
following options to use a response file to configure MicroStrategy:

- **To use a response file with the Configuration Wizard, page 407:**
  This covers the standard procedure of running a response file with
the Configuration Wizard interface.

- To use a response file through the Windows command line, page 408: This covers the procedure of running a response file from the Windows command line. This enables users to run the file without using any graphical user interfaces.

If you are configuring a MicroStrategy installation on Linux, you can use a command line version of the Configuration Wizard to create and use a response file. For steps to create and use a response file as well as perform other configurations using command line tools in Linux, see the Configuring MicroStrategy with a response.ini file, page 703 section in Chapter 12, Configuring MicroStrategy Using Command Line Tools.

- Configuring MicroStrategy components with System Manager: You can use a Configuration Wizard response file as part of an System Manager workflow. System Manager lets you define multiple configurations for your MicroStrategy environment that can be executed in a single workflow. For information on using MicroStrategy System Manager to configure and deploy your MicroStrategy environments, see the System Administration Guide.

To use a response file with the Configuration Wizard

1. Open the MicroStrategy Configuration Wizard. To do this, see To configure MicroStrategy through the Configuration Wizard, page 373.

2. Click Load. The Open dialog box displays.

3. Browse to the path where the response file is saved and click Open. The Summary page opens.

4. An overview of all of the configuration tasks performed by the response file is displayed. Review the configuration tasks and click Finish to perform the configuration. The summary
information is updated as the configurations are completed, providing a way to track the progress of the configurations.

To use a response file through the Windows command line

The steps below are specific to configuring MicroStrategy installed on Windows. For steps to create and use a response file as well as perform other configurations using command line tools in Linux, see Chapter 12, Configuring MicroStrategy Using Command Line Tools.

1. Type the following command in the Windows command line:
   
   macfgwiz.exe -r "Path\response.ini"

   Where Path\ is the fully qualified path to the response file. For example, the common location of a response file is:

   ```
   C:\Program Files\Common Files\MicroStrategy\RESPONSE.INI
   ```

2. If an error message is displayed, check the path and name you supplied for the response file and make any required changes. Repeat the previous step to execute the configuration.

Response configuration parameters and options

It is recommended that you always create the response file through the GUI mode of the Configuration Wizard. However, you can also modify a response file with a text editor to make minor changes such as entering different user login and password information.

⚠ The file must be saved with ANSI encoding.

The response file for configuring MicroStrategy is divided into three areas of configuration, which are described in the sections below:
Creating metadata, History List, and statistics repositories

The response file parameters within the [Repository] section define how metadata, History List, and statistics and Enterprise Manager repositories are created. The table below lists the available parameters and the functionality of available options for each parameter.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Repository]</td>
<td>This section configures the metadata repository and statistics tables. You can have more than one [Repository] section. Additional repository sections can be included as [Repository1], [Repository2], and so on.</td>
</tr>
<tr>
<td>Repository=</td>
<td>Defines whether a metadata, History List, and statistics repositories are configured, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Configures metadata, History List, and statistics repositories.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not configure metadata, History List, and statistics repositories.</td>
</tr>
<tr>
<td>CreateMDTtables=</td>
<td>Defines whether metadata tables are created in a metadata repository, as described below:</td>
</tr>
<tr>
<td></td>
<td>• 1: Creates metadata tables in the metadata repository and creates a default configuration</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not create metadata tables in a metadata repository</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CreateHistListTables=</td>
<td>Defines whether a History List repository is created, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Creates a History List repository.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not create a History List repository.</td>
</tr>
<tr>
<td>CreateStatTables=</td>
<td>Defines whether statistics and Enterprise Manager repositories are created, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Creates statistics and Enterprise Manager repositories.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not create statistics and Enterprise Manager repositories.</td>
</tr>
<tr>
<td>MetadataPath=</td>
<td>Locates the SQL scripts for creating the metadata tables. Example paths to SQL scripts in different environments are listed below:</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environment: C:\Program Files (x86)\Common Files\MicroStrategy\MD8SQL8.sql.</td>
</tr>
<tr>
<td></td>
<td>• Linux: /INTELLIGENCE_SERVER_INSTALL_PATH/mdsql.sql.</td>
</tr>
<tr>
<td>HistoryListPath=</td>
<td>Locates the SQL scripts for creating the History List repository. Example paths to SQL scripts in different environments are listed below:</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environment: C:\Program Files (x86)\Common Files\MicroStrategy\content_server_db_Oracle.sql.</td>
</tr>
<tr>
<td></td>
<td>• Linux: /INTELLIGENCE_SERVER_INSTALL_PATH/content_server_db_Oracle.sql.</td>
</tr>
<tr>
<td>StatisticsPath=</td>
<td>Locates the SQL scripts for creating the statistics and Enterprise Manager repositories. Example paths to SQL scripts in different environments are listed below:</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environment: C:\Program Files (x86)\Common Files\MicroStrategy\StatisticsEnterpriseManagerScripts\DDLScripts\CreateTablesScript.sql</td>
</tr>
<tr>
<td></td>
<td>• Linux: /INTELLIGENCE_SERVER_INSTALL_PATH/statistics_DB2.sql.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>DSNName</td>
<td>Defines the Data Source Name for configuring a metadata repository in the ODBC database.</td>
</tr>
<tr>
<td>UserName</td>
<td>Defines the user name to log in to the database containing the metadata repository.</td>
</tr>
<tr>
<td>UserPwd</td>
<td>Defines the password to log in to the database containing the metadata repository.</td>
</tr>
<tr>
<td>DSNNameHist</td>
<td>Defines the Data Source Name for configuring the History List repository in the ODBC database.</td>
</tr>
<tr>
<td>UserNameHist</td>
<td>Defines the user name to log in to the database for configuring the History List repository.</td>
</tr>
<tr>
<td>UserPwdHist</td>
<td>Defines the password to log in to the database for configuring the History List repository.</td>
</tr>
<tr>
<td>DSNNameStats</td>
<td>Defines the Data Source Name for configuring the statistics and Enterprise Manager repositories in the ODBC database.</td>
</tr>
<tr>
<td>UserNameStats</td>
<td>Defines the user name to log in to the database for configuring the statistics and Enterprise Manager repositories.</td>
</tr>
<tr>
<td>UserPwdStats</td>
<td>Defines the password to log in to the database for configuring the statistics and Enterprise Manager repositories.</td>
</tr>
<tr>
<td>EncryptPassword=</td>
<td>Defines whether the password is encrypted in the response file, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 0: The password is not encrypted in the response file, which enables you to modify the password in the response file later using a text editor. You can then distribute the response file to multiple users with various login and password credentials. However, be aware that this can compromise your database security if you do not remove the password from the response file before distributing it.</td>
</tr>
<tr>
<td></td>
<td>• 1: Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</td>
</tr>
<tr>
<td>DBName=</td>
<td>Defines the database name to create tables in DB2 z/OS. This option should only be used when connecting to a DB2 z/OS database.</td>
</tr>
</tbody>
</table>
Setting up MicroStrategy Intelligence Server

The response file parameters within the [Server] section configures an Intelligence Server definition. The table below lists the available parameters and the functionality of available options for each parameter.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Server]</td>
<td>In this section you can configure the Intelligence Server definition. You can have more than one [Server] section. Additional server sections can be included as [Server1], [Server2], and so on.</td>
</tr>
<tr>
<td>Server=</td>
<td>Defines whether MicroStrategy Intelligence Server is configured, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Configures MicroStrategy Intelligence Server</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not configure MicroStrategy Intelligence Server</td>
</tr>
<tr>
<td>Action=</td>
<td>Defines whether a server definition is created, used, or deleted, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Creates a new server definition</td>
</tr>
<tr>
<td></td>
<td>• 2: Uses an existing server definition</td>
</tr>
<tr>
<td></td>
<td>• 3: Deletes an existing server definition</td>
</tr>
<tr>
<td></td>
<td>• 4: Creates a new server definition and uses it as the default</td>
</tr>
<tr>
<td>InstanceName=</td>
<td>Defines the name of the Intelligence Server</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>instance.</td>
</tr>
<tr>
<td></td>
<td>If you select to delete Intelligence Server instances, you can delete multiple instances by listing multiple instance names, separating each name with the \ character. For example, InstanceName=ServerInstance1\ServerInstance2.</td>
</tr>
<tr>
<td>ProjectsToRegister=</td>
<td>Defines projects to be loaded when Intelligence Server is started. You can select to load multiple projects, separating projects by the \ character. For example, ProjectsToRegister=Project1\Project2.</td>
</tr>
<tr>
<td>ProjectsToUnRegister=</td>
<td>Defines projects to not be loaded when Intelligence Server is started. You can select to not load multiple projects, separating projects by the \ character. For example, ProjectsToUnRegister=Project1\Project2.</td>
</tr>
<tr>
<td>DSName=</td>
<td>Defines the data source name for configuring the MicroStrategy Intelligence Server. This is the data source that stores the metadata.</td>
</tr>
<tr>
<td>DSNUser=</td>
<td>Defines the user name to log in to the metadata database.</td>
</tr>
<tr>
<td>DSNPwd=</td>
<td>Defines the password to log in to the metadata database.</td>
</tr>
<tr>
<td>EncryptPassword=</td>
<td>Defines whether the password is encrypted in the response file, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 0: The password is not encrypted in the response file, which enables you to modify the password in the response file later using a text editor. You can then distribute the response file to multiple users with various login and password credentials.</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>However, be aware that this can compromise your database security if you do not remove the password from the response file before distributing it.</td>
</tr>
<tr>
<td></td>
<td>• 1: Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</td>
</tr>
<tr>
<td>DSSUser=</td>
<td>Defines the MicroStrategy user name to log in to the project.</td>
</tr>
<tr>
<td>DSSPwd=</td>
<td>Defines the password for the MicroStrategy user name to log in to the project.</td>
</tr>
<tr>
<td>MDPrefix=</td>
<td>Defines a prefix for metadata repository tables used by the server definition.</td>
</tr>
<tr>
<td>UseAsDefault=</td>
<td>Defines whether the Intelligence Server definition is set as the default server definition to use for Intelligence Server, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• True: Defines the Intelligence Server definition as the default server definition</td>
</tr>
<tr>
<td></td>
<td>• False: Does not define the Intelligence Server definition as the default server definition</td>
</tr>
<tr>
<td>Port=</td>
<td>Defines the port used by the Intelligence Server. By default, the port is 34952.</td>
</tr>
<tr>
<td>RestPort=</td>
<td>Defines the port used by the REST API Server. By default, the port is 34962.</td>
</tr>
<tr>
<td>RegisterAsService=</td>
<td>This option is only available on Intelligence Servers running on Linux operating systems.</td>
</tr>
<tr>
<td></td>
<td>Defines whether Intelligence Server is registered as a service. Registering Intelligence Server as a service is determined by the following values:</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StartServerAfter Config=</td>
<td>Defines whether Intelligence Server is started after applying the configuration, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Intelligence Server is started after successfully applying the configuration.</td>
</tr>
<tr>
<td></td>
<td>• 0: Intelligence Server is not started after applying the configuration.</td>
</tr>
<tr>
<td>ConfigureSSL=</td>
<td>Defines whether to enable Intelligence Server and Developer to communicate using the SSL protocol, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Enables the use of the SSL protocol for Intelligence Server and Developer communications.</td>
</tr>
<tr>
<td></td>
<td>• 0: Disables the use of the SSL protocol for Intelligence Server and Developer communications.</td>
</tr>
<tr>
<td>SSLPort=</td>
<td>Defines the port to use for SSL access. By default, the port is 39321.</td>
</tr>
<tr>
<td>CertificatePath=</td>
<td>Locates the SSL certificate file you created for Intelligence Server. Type the full path to the SSL certificate file.</td>
</tr>
<tr>
<td>KeyPath=</td>
<td>Locates private key file you created while requesting the certificate for Intelligence Server. Type the full path to the private key file.</td>
</tr>
<tr>
<td>KeyPassword=</td>
<td>Defines the password that you used while creating the key.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DefaultStatisticsRep</td>
<td>Specifies whether you can create a default statistics database instance for all of the projects of the local Intelligence Server metadata, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: You can create a default statistics database instance, using the statistics parameters listed in this table below.</td>
</tr>
<tr>
<td></td>
<td>• 0: A default statistics database instance is not created.</td>
</tr>
<tr>
<td>DefaultDSNNameDefaultStatistics</td>
<td>Specifies the data source name for your statistics repository.</td>
</tr>
<tr>
<td>UserNameDefaultStatistics</td>
<td>Specifies the database user name for the user that can connect to the statistics data source.</td>
</tr>
<tr>
<td>UserPwdDefaultStatistics</td>
<td>Specifies the password for the user that can connect to the statistics data source.</td>
</tr>
<tr>
<td>EncryptUserPwdDefaultStatistics</td>
<td>Defines whether the statistics user password is encrypted in the response file, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 0: The password is not encrypted in the response file, which enables you to modify the password in the response file later using a text editor. You can then distribute the response file to multiple users with various login and password credentials. However, be aware that this can compromise your database security if you do not remove the password from the response file before distributing it.</td>
</tr>
<tr>
<td></td>
<td>• 1: Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</td>
</tr>
</tbody>
</table>
### Options Description

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DefaultStatisticsPrefix</td>
<td>Defines a prefix for statistics repository tables used by the server definition.</td>
</tr>
<tr>
<td>kafkaHost(s)=</td>
<td>Provide host name or IP address of Messaging Services. Format for this setting should be: server1.acme.com:9092, server2.acme.com:9093,…</td>
</tr>
<tr>
<td>ConfigMessagingService=</td>
<td>Enable or disable Messaging Services for Intelligence Server and configure Messaging Services host and port settings.</td>
</tr>
</tbody>
</table>

### Creating and configuring project sources

The response file parameters within the **[Client]** section create and configure project sources. The table below lists the available parameters and the functionality of available options for each parameter.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[Client]</strong></td>
<td>In this section you can configure the project source name. You can have more than one <strong>[Client]</strong> section. Additional client sections can be included as <strong>[Client1]</strong>, <strong>[Client2]</strong>, and so on.</td>
</tr>
<tr>
<td>Client=</td>
<td>Defines whether project sources are configured, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Configures project sources</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not configure project sources</td>
</tr>
<tr>
<td>EncryptPassword=</td>
<td>Defines whether the password is encrypted in the response file, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 0: The password is not encrypted in the response file, which enables you to modify the password in the response file later using a text editor. You can then distribute the response file</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>to multiple users with various login and password credentials. However, be aware that this can compromise your database security if you do not remove the password from the response file before distributing it.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>• 1: Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DataSource=</strong></td>
<td>Defines the name of the new project source to create.</td>
</tr>
<tr>
<td><strong>ConnType=</strong></td>
<td>Defines the database connection type for a project source. The following connection types are supported:</td>
</tr>
<tr>
<td><strong>• 2: Connects a project source to the metadata using an ODBC DSN (Windows only).</strong></td>
<td></td>
</tr>
<tr>
<td><strong>• 3: Connects a project source to the metadata through a MicroStrategy Intelligence Server (three-tier).</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DSN=</strong></td>
<td>If using connection type 2 (ConnType=2), defines the name of the ODBC database.</td>
</tr>
<tr>
<td><strong>UserName=</strong></td>
<td>If using connection type 2 (ConnType=2), defines the user name to connect to the ODBC database.</td>
</tr>
<tr>
<td><strong>UserPwd=</strong></td>
<td>If using connection type 2 (ConnType=2), defines the password to log in to the database.</td>
</tr>
<tr>
<td><strong>ServerName=</strong></td>
<td>If using connection type 3 (ConnType=3), defines the name of the MicroStrategy Intelligence Server to connect to.</td>
</tr>
<tr>
<td><strong>Port=</strong></td>
<td>If using connection type 3 (ConnType=3), defines the port number for the Intelligence Server when creating a server (three-tier) project source. The default port number for MicroStrategy Intelligence Server is 34952.</td>
</tr>
<tr>
<td><strong>Authentication=</strong></td>
<td>The following authentication modes are supported:</td>
</tr>
<tr>
<td><strong>• 1: Standard or login ID and password entered by the user</strong></td>
<td></td>
</tr>
<tr>
<td><strong>• 2: Network login ID (Windows authentication)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>• 8: Guest account (Anonymous authentication)</strong></td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>• 16: LDAP authentication</td>
<td></td>
</tr>
<tr>
<td>• 32: Database login ID and password (database authentication)</td>
<td></td>
</tr>
<tr>
<td>• 128: Integrated authentication</td>
<td></td>
</tr>
</tbody>
</table>

For information on the available authentication modes, see the *Authentication modes, page 402.*

**MDPrefix=**

If using connection type 2 (ConnType=2), defines a prefix for metadata repository tables.

**Timeout=**

Defines and enforce a connection time out for inactive users connected to a project source. The following values are supported:

• 0: Defines that users are not disconnected from project sources due to inactivity.

• Numerical value greater than 0: A numerical value (in minutes) greater than 0 defines the amount of inactivity that is allowed before a user is automatically disconnected from a project source.

---

**Creating and Configuring the Enterprise Manager Project**

The parameters in the `[EMProjectHeader]` portion of the response file create the Enterprise Manager project on this machine and configure the connection to the Statistics and Enterprise Manager repository. The table below lists the available parameters and the functionality of available options for each parameter. For detailed information about each parameter, see the Configuration Wizard Help.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>[EMProjectHeader]</code></td>
<td>Options in this portion refer to creating the Enterprise Manager project on this machine.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EMProject=</td>
<td>Defines whether to create the Enterprise Manager project, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Create Enterprise Manager project on this machine.</td>
</tr>
<tr>
<td></td>
<td>• 0: Do not create the Enterprise Manager project.</td>
</tr>
<tr>
<td>EMProjectEncryptPwd=</td>
<td>Defines whether the passwords are encrypted in the response file, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 0: The passwords are not encrypted in the response file, which enables you to modify the passwords in the response file using a text editor. You can then distribute the response file to multiple users with various login and password credentials. However, be aware that this can compromise your database security if you do not remove the passwords from the response file before distributing it.</td>
</tr>
<tr>
<td></td>
<td>• 1: Encrypts the passwords in the response file, which ensures that your passwords are secure. This is the default behavior.</td>
</tr>
<tr>
<td>EMProjectDSSUser=</td>
<td>The user name to log in to the Enterprise Manager project.</td>
</tr>
<tr>
<td>EMProjectDSSPwd=</td>
<td>The password for the user name above. This may be encrypted, depending on the EMProjectEncryptPwd= setting.</td>
</tr>
<tr>
<td>EMProjectPkgFile=</td>
<td>The full path and file name of the MicroStrategy Enterprise Manager project package file used to create the project. On Windows, by default this is C:\Program Files (x86)\Common Files\MicroStrategy\OOTB-EM.mmp.</td>
</tr>
<tr>
<td>EMProjectDSNName=</td>
<td>The Data Source Name for the database that contains your Statistics and Enterprise Manager repository.</td>
</tr>
</tbody>
</table>
### Options Description

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>EMProjectDSNUserName=</code></td>
<td>The user name to connect to the Statistics and Enterprise Manager repository database.</td>
</tr>
<tr>
<td><code>EMProjectDSNUserPwd=</code></td>
<td>The password for the user name above for the Statistics and Enterprise Manager repository database. This may be encrypted, depending on the <code>EMProjectEncryptPwd=</code> setting.</td>
</tr>
</tbody>
</table>

### Creating and Configuring the Platform Analytics Project

The parameters in the `[PAProjectHeader]` portion of the response file create the Platform Analytics project on this machine and configure the connection to the Platform Analytics repository. The table below lists the available parameters and the functionality of available options for each parameter. For detailed information about each parameter, see the Configuration Wizard Help.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>[PAProjectHeader]</code></td>
<td>Options in this portion refer to creating the Platform Analytics project on this machine.</td>
</tr>
<tr>
<td><code>PAProject=</code></td>
<td>Defines whether to create the Platform Analytics project, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>- 1: Create Platform Analytics project on this machine.</td>
</tr>
<tr>
<td></td>
<td>- 0: Do not create the Platform Analytics project.</td>
</tr>
<tr>
<td><code>PAProjectEncryptPwd=</code></td>
<td>Defines whether the passwords are encrypted in the response file, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>- 0: The passwords are not encrypted in the response file, which enables you to modify the passwords in the response file using a text editor. You can then distribute the response file to multiple users with various login and password credentials. However, be aware that this can</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>compromease your database security if you do not remove the passwords from the response file before distributing it.</td>
<td></td>
</tr>
<tr>
<td>1: Encrypts the passwords in the response file, which ensures that your passwords are secure. This is the default behavior.</td>
<td></td>
</tr>
<tr>
<td>PAPROJECTDSSUser</td>
<td>The user name to log in to the Platform Analytics project.</td>
</tr>
<tr>
<td>PAPROJECTDSSPassword</td>
<td>The password for the user name above. This may be encrypted, depending on the PAProjectEncryptPassword setting.</td>
</tr>
<tr>
<td>PAProjectProjectPackageFile</td>
<td>The full path and file name of the MicroStrategy Platform Analytics project package file used to create the project.</td>
</tr>
<tr>
<td>On Windows, by default this is C:\Program Files (x86)\Common Files\MicroStrategy\PlatformAnalyticsProjectObjects.mmp.</td>
<td></td>
</tr>
<tr>
<td>PAConfigurePackageFile</td>
<td>The full path and file name of the MicroStrategy Platform Analytics project configuration package file used to create the project.</td>
</tr>
<tr>
<td>On Windows, by default this is C:\Program Files (x86)\Common Files\MicroStrategy\PlatformAnalyticsConfigurationObjects.mmp.</td>
<td></td>
</tr>
<tr>
<td>PAPROJECTDSSName</td>
<td>The Data Source Name for the database that contains your Platform Analytics repository.</td>
</tr>
<tr>
<td>PAPROJECTDSUsername</td>
<td>The user name to connect to the Platform Analytics repository database.</td>
</tr>
<tr>
<td>PAPROJECTDSPassword</td>
<td>The password for the user name above for the Platform Analytics repository database. This may be encrypted, depending on the PAProjectEncryptPassword setting.</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAProjectDSNPrefix</td>
<td>The prefix for the Platform Analytics repository tables.</td>
</tr>
</tbody>
</table>

### Encryption Key Manager

Encryption Key Manager (EKM) creates and manages unique encryption keys for every MicroStrategy environment. The EKM features include creating, importing, and exporting of these unique keys through Configuration Wizard. These keys encrypt potentially sensitive information stored in the metadata, cube, cache, history list, and session recovery files.

### Terms and Definitions

- **Master Key**: The master key encrypts the key store and is saved in the master key file. MicroStrategy Intelligence Server will look for the path to the master key in the registry upon start up.

- **Key Store**: Contains keys used to encrypt the metadata and file caches. These keys are encrypted by the master key.

- **Secure Bundle**: A password protected file that enables administrators to securely deploy encryption keys between clustered Intelligence Servers or servers sharing the same metadata.

- **Secure Bundle Code**: The password used to protect the Secure Bundle file.
High Level Steps to Use the Encryption Key Manager

1. Enable the Encryption Key Manager Feature and restart Intelligence Server.

2. Using Configuration Wizard:
   - Update the metadata to apply the new encryption keys.
   - Export the Secure Bundle.

3. To configure additional nodes in a clustered environment:
   - Enable the Encryption Key Manager Feature and restart Intelligence Server.
   - Import the Secure Bundle to each node using Configuration Wizard.

Enable the Encryption Key Manager Feature

ℹ️ The Encryption Key Manager is disabled by default.

Windows

To enable the Encryption Key Manager:

1. Open the Registry Editor


3. Double click KE/EncryptionKeyManager.

4. Change the Value Data field from 0 to 1.

5. Click OK.

6. Restart Intelligence Server.

ℹ️ If Configuration Wizard is open it must be restarted as well.
Linux

To enable the Encryption Key Manager:

1. Locate the MSIReg.reg file in your MicroStrategy root install directory.

2. Modify the following in a text editor:

   Change

   
   \[HKEY_LOCAL_MACHINE\SOFTWARE\MicroStrategy\Feature Flags\]
   "KE/EncryptionKeyManager"=dword:00000000

   to

   \[HKEY_LOCAL_MACHINE\SOFTWARE\MicroStrategy\Feature Flags\]
   "KE/EncryptionKeyManager"=dword:00000001

3. Save and close.

4. Restart Intelligence Server.

   If Configuration Wizard is open it must be restarted as well.

Updating the Metadata with Encryption Key Manager

Once Encryption Key Manager is enabled the metadata must be updated to become encrypted. The encryption keys and master key are automatically generated and stored locally during the metadata upgrade. See the Update the Metadata chapter in the Upgrade Guide for steps to complete this process.
Managing Encryption Key Configuration Options

The import and export features of EKM allow you to securely share and transmit encryption keys using a secure bundle.

Importing and Exporting Encryption Keys to a Secure Bundle

Clustered Intelligence Servers or servers sharing the same metadata must use the same set for encryption keys. This is accomplished by exporting the secure bundle from the first configured Intelligence Server and then importing the secure bundle to the other servers.

For recovery purposes you should export and store a secure bundle outside of your MicroStrategy environment, even if you are only using a single Intelligence Server instance. You will not be able to recover the information stored in your metadata without the secure bundle and secure bundle code.

Exporting to a Secure Bundle

1. Open Configuration Wizard.
2. Select Manage Encryption Keys.
3. Click Next.
4. Select Export encryption keys to a Secure Bundle.
5. Click Next.
6. Enter your Administrator user name and password.
7. Click Next.
8. Accept the auto-generated Secure Bundle Code or click the refresh button to regenerate the code.

Record the generated code value. It will be needed for the importing process.
9. Click **Next**.

10. Click **Finish** to export the Secure Bundle.

11. Review the message on the warning dialog box and click **OK**.

Importing a Secure Bundle

You can use this option to configure additional nodes in a clustered environment. Ensure that you have updated the metadata associated with the Intelligence Server before importing the Secure Bundle.

1. Copy the exported Secure Bundle to the target Intelligence Server.

2. Open Configuration Wizard.

3. Select **Manage Encryption Keys**.

4. Click **Next**.

5. Select **Import encryption keys from a Secure Bundle**

6. Click **Next**.

7. Enter your Administrator user name and password.

8. Click **Next**.

9. Click the browse button and navigate to the Secure Bundle location.

10. Click Open to select the file.

11. Enter the Secure Bundle Code from the export process.

12. Click **Next**.

13. Click **Finish** to import the Secure Bundle.

14. Review the message on the warning dialog box and click **OK**.
Generating a New Master Key

1. Copy the exported Secure Bundle to the target Intelligence Server.
2. Open Configuration Wizard.
3. Select Manage Encryption Keys.
4. Click Next.
5. Select Generate New Master Key.
6. Click Next.
7. Enter your Administrator user name and password.
8. Click Next.
9. The location of the current Master Key and the destination for the new Master Key are displayed.
10. Click Next.
11. Click Finish to generate the new Master Key.

Changing the Master Key Location

To modify the location where the Intelligence Server looks for the Master Key file:

1. Copy the exported Secure Bundle to the target Intelligence Server.
2. Open Configuration Wizard.
3. Select Manage Encryption Keys.
4. Click Next.
5. Select Change Master Key Location.
6. Click Next.
7. Enter your Administrator user name and password.
8. Click **Next**.
9. Click the browse button and navigate to the new location for the Master Key.
10. Specify a name for the Master Key file and click **Save**.
11. Click **Finish** to change the Masker Key location.
12. Review the message on the warning dialog box and click **OK**.

### Connecting to a Data Warehouse and Other Repositories

For MicroStrategy users to be able to browse attribute elements and execute reports, a connection to a data warehouse must be created. A connection to other data sources can also support History Lists, statistics, and including data from multiple data sources into your MicroStrategy project.

You can perform data source connection tasks from the Project Configuration Editor, which can be accessed by right-clicking a project and selecting **Project Configuration**.

Information: The tasks described in this section require MicroStrategy Administrator privileges.

### Specifying Warehouse Connection Information

A database instance is a MicroStrategy object, created in MicroStrategy Developer by an administrator, that represents a connection to a data source. A database instance specifies connection information, such as the data source name, Login ID and password, and other data source specific information.
The steps to create the required components of a database instance are provided in the following sections: Creating a database instance, page 435, Creating a database connection, page 439, and Creating a database login, page 445.

When a project architect creates a project, the architect assigns a database instance to that project. A project specifies only one warehouse database instance at a time, but a database instance can be assigned to multiple projects. Since only one data source can be included in the project's relational schema, all reports and documents return information from a single data source.

If you have a license for the MultiSource Option feature, you can connect a project to multiple warehouse database instances. There can be multiple data sources that connect to the Warehouse Catalog for the project. Since these data source can be integrated as part of the project's relational schema, all reports and documents can return information from multiple data sources. For information on accessing multiple data sources in a project, see the Project Design Guide.

Regardless of whether you have a license for the MultiSource Option, you can also extend a project's access to multiple data sources through other MicroStrategy features. Freeform SQL, Query Builder, and supporting access through MicroStrategy to other MDX cube sources such as SAP BW, Oracle Essbase, and Microsoft Analysis Services allows non-project database instances to be included and used in a project along with the warehouse database instances. For information on Freeform SQL and Query Builder, see the Advanced Reporting Guide. For information on MDX cube sources, see the MDX Cube Reporting Guide.

These non-project database instances can allow a project to connect to the data sources for the various features and additional data sources mentioned above, instead of accessing the data from the
project's relational schema. For more information on the Warehouse Catalog, see the Project Design Guide.

SQL data warehouses database instances

A SQL data warehouse database instance is any database instance that connects to a database or other data source through SQL queries. More specifically, this covers database instances used for standard MicroStrategy reporting, Freeform SQL, Query Builder, data marts, and any other relational data source. You can also connect to History List and statistics tables through SQL data warehouse database instances. The SQL data warehouse database instances are available in the Project Configuration Editor, as shown below.

Selecting a database instance check box makes that database instance available in the project for standard MicroStrategy reporting, data marts, Query Builder, and Freeform SQL. If you have a license for the MultiSource Option, selecting a check box for a database instance also makes the database instance available from the Warehouse Catalog to be part of the project's relational schema.
Database instances can be created as part of the Import Data feature. A database instance used for the Import Data feature is displayed with the icon. These database instances are created with security permissions for the user that created them while using the Import Data feature. If you select one of these database instances to be included as an available database instance in the project, it is recommended that you change the security permissions to a MicroStrategy user with administrative privileges. This includes taking ownership of the database instance and defining an appropriate access control list. This ensures that no changes are made to the database instance by other users, which could cause a loss of connectivity to the data source. For information on the Import Data feature, refer to the MicroStrategy Web online help.

The shading and color of a database instance in the list of relational database instances reflects how the database instance is being used in the project:

- **Blue text:** This is the warehouse database instance, as selected from the warehouse database instance drop-down list. There can only be one warehouse database instance for a project, because this database instance's data is populated in the Warehouse Catalog to define the project's relational schema. You cannot choose to disable the warehouse database instance for the project without first selecting a different warehouse database instance.

If you have a license for the MultiSource Option, the primary database instance acts as the main source of data for a project and is used as the default database instance for tables added to the project.

For information on the Warehouse Catalog as well as accessing multiple data sources with the MultiSource Option, see the Project Design Guide.
- **Bold text**: The project contains objects that are dependent on the database instance. You cannot choose to disable a database instance that has dependent objects for the project.

- **Normal text**: The database instance is not being used in the project.

Clearing the check box of a database instance removes the database instance from the project and deletes any unused Freeform SQL or Query Builder schemas. You can clear a database instance from a project only if there are no dependent objects in the project for the database instance. For more information on removing a database instance and related Freeform SQL and Query Builder schemas from a project, refer to the System Administration Guide.

**MDX cube database instances**

An MDX cube database instance is any database instance that connects to an MDX cube source, such as SAP BW, Oracle Essbase, or Microsoft Analysis Services. For information on connecting to and reporting on these MDX cube sources, refer to the MDX Cube Reporting Guide. The MDX cube database instances are available in the Project Configuration Editor, as shown below.
A database instance that has an MDX cube schema is represented with bold text. The shading and color of a database instance in the list of relational database instances reflects how the database instance is being used in the project:

- **Bold**: The project contains objects that are dependent on the database instance. You cannot choose to disable a database instance that has dependent objects for the project.

- **Normal**: The database instance is not being used in the project.

If you remove an MDX cube database instance from a project, you can delete any unused MDX cube schema objects. You can remove database instance from a project only if there are no dependent objects in the project for the database instance. For more information on removing a database instance and related MDX cube managed objects from a project, refer to the System Administration Guide.

For additional information on configuring MDX cube database instances, refer to the MDX Cube Reporting Guide.
MDX schema loading and maintenance

You can click Schema Maintenance to perform various tasks for an MDX cube schema that is part of your project, as described below:

- You can choose when an MDX cube schema associated with a database instance is loaded for a project. By default, MDX cube schemas are loaded as needed when MDX cube reports are executed. You can also choose to load MDX cube schemas when Intelligence Server starts. For information on defining when MDX cube schemas should be loaded, refer to the MDX Cube Reporting Guide.

- When you integrate MDX cube sources into MicroStrategy, the data is integrated as an MDX cube schema. Once you integrate an MDX cube source into MicroStrategy, you can exchange the database instance used to connect to the MDX cube schema for a different database instance. This allows you to use different database instances with different login and connection information to access an MDX cube schema. For information on exchanging the database instance used to connect to the MDX cube schema, refer to the MDX Cube Reporting Guide.

Creating a database instance

Database instances are created and modified in the Database Instance Manager, which can be found by expanding Administration for a project source, then expanding Configuration Managers. When you choose to create a new database instance, the Database Instances Editor opens.
You can also create a new database instance using the Database Instance Wizard that is available in the Database Instance Manager shortcut menu.

The Database Instances Editor has the following tabs:

- **General**—specifies the database instance name, connection type (data source platform or applicable data source), and default database connection.

  The database connection type you choose should match your data source and determines whether the database instance is a relational or an MDX cube database instance.

- **Advanced**—specifies the database name for intermediate table storage if a database other than the warehouse is used to store intermediate tables, as well as other options.
The Advanced tab is not available for MDX cube database instances.

- Job Prioritization—specifies the job prioritization scheme for the instance and the number of prioritized connections.

To create a database instance

1. In MicroStrategy Developer, log in to a project source with administrative privileges.
2. Go to Administration > Configuration Managers > Database Instances.
3. From the File menu, go to New > Database Instance. The Database Instances Editor opens.
4. On the General tab, in the Database instance name field, type the name of the database instance.
5. In the Database connection type drop-down list, select the data source connection type according to the data source hosting your database.

   If you have upgraded from a previous version of MicroStrategy, you can click Upgrade to retrieve any database connection types that have been included since the previous version of MicroStrategy that you used.

6. On the Advanced tab, you can configure various options for the database instance, including:
   - Intermediate table storage: You can specify the database name and table name space to use when intermediate tables are created. Intermediate tables are created to support various queries.
• **Database gateway support**: You can support backwards compatibility for database gateway support from MicroStrategy version 6.x.

  To enable database gateway support, select the **Primary database instance** check box, and then select a primary database instance from the drop-down list. The primary database instance is the database instance that should be used for element browsing against the selected table and for queries that do not require joins to other tables. For information on database gateway support, see the Project Design Guide.

• **Data mart optimization**: You can support data mart optimization if the data source for the database instance is in the same data source that contains data marts.

  To enable data mart optimization, select the **This database instance is located in the same warehouse as** check box, and then select a database instance from the drop-down list.

• **Table prefix**: If the tables in your data source use a table prefix, you can include the table prefix to identify the proper collection of tables. Click **Select** to select a table prefix or define a new table prefix.

• **ODBC Version**: You can define which ODBC version to use for the database instance, as described below:

  ■ **Use 2.0 ODBC Calls**: ODBC 2.0 was used in pre-9.0 versions of MicroStrategy. You can use this option for backward compatibility if your database management system does not support ODBC 3.x. This also allows you to use extended fetch to retrieve blocks of data from the database into memory, instead of row by row, which is included in the
steps *Specifying Warehouse Connection Information, page 429.*

- **Use 3.x ODBC Calls:** The support of ODBC 3.x is introduced in MicroStrategy 9.0. This value is chosen by default for Database Instances in MicroStrategy 9.x and higher. You should use this option if your database management system supports ODBC 3.x. ODBC 3.x will always use extended fetch (SQLFetchScroll) to retrieve blocks of data from the database into memory rather than row-by-row retrieval.

7. On the **Job Prioritization** tab, you can configure how jobs are prioritized for the database instance. For information on configuring job prioritization, see the *System Administration Guide.*

8. On the **General** tab, in the **Database connection (default)** pane, select the default data source connection and click **OK.**

   If the necessary database connection does not exist, you can create one by clicking **New.** For steps to create a database connection, see *Creating a database connection, page 439* below.

### Creating a database connection

A database connection specifies the DSN and database login used to access the data source. A database instance designates one database connection as the default connection for MicroStrategy users; however, users and groups can be mapped to other database connections using connection mapping. For more details on connection mapping, see *User connection mapping, page 447.*

You create database connections in the Database Instances Editor by clicking **New** on the General tab. Any database connection created within the Database Instances Editor is available for use across all database instances in the project source. For more information on
creating a database connection for MDX cube sources, refer to the 
**MDX Cube Reporting Guide.**

When you choose to create a new database connection, the Database Connections dialog box opens:

![Database Connections Dialog Box](image)

The Database Connections dialog box has different options depending on the database instance type:

- **SQL data warehouse database instances**
  - **General:** Specifies the database connection name, the warehouse DSN, and the default database login.
  - **Advanced:** Specifies the database driver mode, driver execution mode, and other miscellaneous warehouse connection settings.
MDX cube database instances

- General: Specifies the database connection name, the default database login, and additional connection information that you must provide. For more information on creating a database connection for MDX cube sources, see the MDX Cube Reporting Guide.

- Advanced: Specifies the connection settings, additional connection string parameters, and connection caching settings.

To create a database connection

Before moving forward, ensure that a database instance has been created as described in Creating a database instance, page 435.

1. On the General tab, in the Database connection name box, type a name to identify the database connection.

2. In the Local system ODBC data sources pane, select the data source name for the data source.

3. On the Advanced tab, you can define various options per your requirements and the requirements of the database you are connecting to, including:

   - Database driver mode: Select one of the following database driver modes:
     - Multi-process: Each connection to the warehouse database is spawned as a separate process, identified in Windows Task Manager as M8DBMPE.exe. If one process fails, such as when a database access thread hangs or is lost, other processes are not affected.
     - Multi-threaded: All connections to the warehouse database are maintained inside the Intelligence Server process
**MSTRSVR.exe.** All connections, SQL submissions, and data retrievals from the database are handled within this process.

MicroStrategy recommends setting all database drivers to multi-process mode. The robustness and stability which come with multi-process mode greatly overshadow any increased efficiency that may come with multi-threaded mode. Problems that appear random and sporadic in multi-threaded operation can often be resolved by switching to multi-process mode.

- **Driver execution mode:** Define the driver execution mode depending on the ODBC driver being used:
  - **Asynchronous Connection:** All statements allocated within the connection should be able to run SQL asynchronously.
  - **Asynchronous Statement:** For each statement, the asynchronous mode is explicitly set.
  - **Synchronous:** Only one statement executes at a time. This is the default value.

Many newer ODBC drivers do not support asynchronous mode because the driver is capable of opening a new thread and executing a new query while simultaneously running an earlier query. The *MicroStrategy Readme* gives recommendations for the driver execution mode options that can be used for different ODBC drivers.

- **Use extended fetch:** Select this check box to enable Intelligence Server to fetch blocks of data from the database into memory, instead of row-by-row. Be aware that this check box is only available if the database instance is defined to use 2.0 ODBC calls, which is included in the steps *Specifying*
Warehouse Connection Information, page 429. When 3.0 ODBC calls are enabled, extended fetch is already enabled and this option is grayed out.

- **Use parameterized queries**: Select this check box to enable Intelligence Server to pass data to the database in blocks instead of row-by-row. For information on how parameterized queries can improve performance in MicroStrategy, see the Project Design Guide.

- **Maximum cancel attempt time (sec)**: Defines the maximum amount of time the MicroStrategy Query Engine waits for a successful attempt before it cancels a query. Values of 0 and -1 indicate no limit.

- **Maximum query execution time (sec)**: Defines the maximum amount of time a single pass of SQL can execute on the database. Values of 0 and -1 indicate no limit.

- **Maximum connection attempt time (sec)**: Defines the maximum amount of time Intelligence Server waits to connect to the database. Values of 0 and -1 indicate no limit.

- **Additional connection string parameters**: Enables you to pass additional ODBC connection parameters to the database as part of the connection string. This is useful if you need to change ODBC defaults. Click **Preview** to see the entire connection string.

- **Table prefix**: Defines a table prefix that specifies the schema containing the tables to access.

- **Character set encoding for Windows drivers**: The options listed below are only relevant when Intelligence Server is running on a Windows machine:
- **Non UTF-8** (default): Select this option if the ODBC driver returns information in a character encoding other than UTF-8.

- **UTF-8**: Select this option if the ODBC driver returns information in UTF-8 character encoding. Drivers for Teradata databases may require UTF-8 encoding.

- **Character set encoding for UNIX drivers**: The options listed below are only relevant when Intelligence Server is running on a UNIX machine:
  - **Non UTF-8**: Select this option if the ODBC driver returns information in a character encoding other than UTF-8.
  - **UTF-8 (default)**: Select this option if the ODBC driver returns information in UTF-8 character encoding. Drivers for Teradata databases may require UTF-8 encoding.

- **Connection Caching**: Specify the caching of the database connection using the following options:
  - **Connection idle timeout (sec)**: Defines the amount of time an inactive connection to the database remains cached until it is terminated. You must also set the Connection lifetime, described above, to a value greater than zero for database connections to be used by more than a single job.

  Enforcement of the connection idle timeout can cause a database connection to be removed before it reaches its connection lifetime. You can use this connection idle timeout to ensure that database connections do not remain in Intelligence Server memory in an idle state for an extended amount of time.

  Enforcement of the connection idle timeout can cause a database connection to be removed before it reaches its
connection lifetime. You can use this connection idle timeout to ensure that database connections do not remain in Intelligence Server memory in an idle state for an extended amount of time.

Enforcement of the connection idle timeout can cause a database connection to be removed before it reaches its connection lifetime. You can use this connection idle timeout to ensure that database connections do not remain in Intelligence Server memory in an idle state for an extended amount of time.

If you type a value of 0, when the job associated with a database connection is completed, the database connection is deleted and not put into a cache. If you type a value of -1, a database connection can remain idle and considered for new jobs until the database connection lifetime is reached.

4. On the General tab, in the Default database login name pane, select the default database login and click OK.

If the necessary database login does not exist, you can create one by clicking New. For steps to create a database connection, see Creating a database login, page 445 below.

Creating a database login

A database login specifies the user ID and password used to access the data source. The database login overwrites any login information stored in the DSN. A database connection designates one database login as the default login for MicroStrategy users, however users and groups can be mapped to other database logins using connection mapping.

Connection mapping is explained in User connection mapping, page 447.
You create database logins in the Database Connections dialog box by clicking **New** on the General tab. Any database login created within the Database Connections dialog box is available for use across all database connections in the project source.

MicroStrategy reporting and analysis features require a general set of database login permissions that can connect to and modify the data source and metadata, as described below:

- For the metadata, the Select, Insert, and Update permissions are required. Intermediate tables are created in the metadata for recursive search queries, which requires Create and Drop permissions as well. Updating the schema requires the Delete permission.

- For the data warehouse, the Select, Create, Insert, and Drop permissions are required.

When you choose to create a new database login, the Database logins dialog box opens:

![Database Logins dialog box](image)

To create a database login

Before moving forward, ensure that the following is complete:

- A database instance has been created, as described in *Creating a database instance, page 435.*
A database connection has been created, as described in Creating a database connection, page 439.

1. In the **Database Login** field, type the name of the database login.

2. Provide the user ID and password required to access the data source, using one of the following methods:
   - Type the user ID in the **Login ID** field, and type the password for that user ID in the **Password** field.
   - Select the **Use network login ID** check box to connect to the data source using the network user credentials which are also used to run Intelligence Server. If Intelligence Server is running as a service, this is the user that is running the `mstrsvr.exe` process. To determine this user, in MicroStrategy Service Manager, select **MicroStrategy Intelligence Server** and click **Options**. The user is listed on the Service Startup tab, in the Login field. If the Service Account Name is defined as System Account, the Windows user credentials are used to access the data source.

3. Click **OK**.

   Database logins are passed to the data source any time a user executes a report or browses attribute elements. Therefore, all database logins created in MicroStrategy Developer must be also be created as valid logins in the data source.

### User connection mapping

User connection mapping is 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.

MicroStrategy users link to database connections and logins using:
The default database connection (and, therefore, default database login)

- Specialized maps to a database connection and/or database login (different than the default connection and login) for either a user or user group

You can map users to connections and logins in the Project Configuration Editor or Command Manager. For information about how connection maps are used, see the System Administration Guide.

MicroStrategy reporting and analysis features require a general set of database login permissions to connect to and modify the data warehouse and metadata, as described below:

- For the metadata, the Select, Insert, and Update permissions are required. Intermediate tables are created in the metadata for recursive search queries, which requires Create and Drop permissions as well. Updating the schema requires the Delete permission.

- For the data warehouse, the Select, Create, Insert, and Drop permissions are required.

To create a connection map

Before moving forward, ensure that the following is complete:

- A database instance has been created, as described in Creating a database instance, page 435.

- A database connection has been created, as described in Creating a database connection, page 439.

- A database login has been created, as described in Creating a database login, page 445.
1. In Developer, log in to a project.

2. Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.

3. In the **Categories** list, expand the **Database Instances** category, and then select **Connection mapping**.

4. Right-click in the **Database instances - Connection mapping** pane, and select **New**. A new connection mapping is added.

5. You can define the connection mapping by specifying the information described below:

   - **Database Instance**: The database instance which connects to the data source required for the connection mapping.
   - **User**: The user or user group to apply the connection mapping to.
   - **Language**: The language of the data accessed by the connection mapping. You can use connection mappings to support data internationalization. For information on supporting data internationalization with connection mappings, see the **Project Design Guide**.
   - **Database connection**: The data source to connect to.
   - **Database Login**: The database login for the connection mapping.

6. Click **OK**.

**Creating a Project**

Now you have configured Intelligence Server and are ready to create a project. There are various ways to create a project to get your MicroStrategy project started. The different methods to create a project are described in the **Project Design Guide**.
The MicroStrategy platform provides a Tutorial project, which is a sample data warehouse and demonstration project you can use to learn about the various features that MicroStrategy offers. It is ready to be used and requires no additional configuration tasks. To use the MicroStrategy Tutorial, refer to the Basic Reporting Guide for more information. To create a new project using your own data, see the Project Design Guide.

Configuring your MicroStrategy Installation

To help guide the rest of your installation and configuration steps, refer to the section Installing and Configuring MicroStrategy on Windows, page 111 in Chapter 1, Planning Your Installation, for an installation and configuration checklist.

Data Import with Proxy

Introduction to Data Import with Proxy

This section provides an overview of all of the capabilities to get data through proxy. You will learn how to best leverage these capabilities to import data and create or update your datasets.

Background

Many enterprise customers cannot import data from URL sources, as they are in a proxy environment. This considerably inhibits their ability to ingest data from external sources, and limits MicroStrategy’s ability to analyze the data. Supporting proxy considerably increases their ability to analyze any third-party data that is currently available via import from URL. There are also customers in a proxy environment who cannot access Facebook, Twitter, or other OAuth sources. In short, supporting proxy configuration is needed for customers in some
limited network environment, which is typical in enterprise deployments.

Currently, we do not support the proxy feature on data from URL. There is partial support in some OAuth sources, such as Twitter or Facebook, but configurations are not unified on different platforms, which may cause confusion. Furthermore, we do not support PAC (Proxy Auto-Config) to configure the proxy.

See also *How to Configure a Proxy*.

**How to Configure a Proxy**

**How to Configure a Proxy in Intelligence Server for Windows**

1. **Find the configuration file.**

   After you install or update Workstation for Windows to 10.11 or later, the default configuration file, `proxy.json`, is installed automatically.

   The file `proxy.json` is located in `C:\Program Files (x86)\Common Files\MicroStrategy for Intelligence Server` by default.

2. **Configure the `proxy.json` file.**

   Open `proxy.json` with a text editor. The following is the default configuration:

   ```json
   {
     "ENABLE_FLAG": "FALSE",
     "USE_PROXY_PAC": "FALSE",
     "PAC_URL": "",
     "USE_PROXY_STATIC": "FALSE",
     "PROXY_HOST": "",
     "PROXY_PORT": "",
     "PROXY_AUTHENTICATION": "FALSE",
   }
   ```
You want to access data through the following authenticated proxy:

- **Host**: 10.0.0.1
- **Port**: 8080
- **Username**: mstr
- **Password**: 123456

Configure `proxy.json` as follows:

```json
{
    "ENABLE_FLAG": "TRUE",
    "USE_PROXY_PAC": "FALSE",
    "PAC_URL": "",
    "USE_PROXY_STATIC": "TRUE",
    "PROXY_HOST": "10.0.0.1",
    "PROXY_PORT": "8080",
    "PROXY_AUTHENTICATION": "TRUE",
    "USER_NAME": "mstr",
    "PASSWORD": "123456"
}
```

If the proxy server is not authenticated, then just set `PROXY_AUTHENTICATION` to `FALSE`. Configure it like this:

```json
{
    "ENABLE_FLAG": "TRUE",
    "USE_PROXY_PAC": "FALSE",
    "PAC_URL": "",
    "USE_PROXY_STATIC": "TRUE",
    "PROXY_HOST": "10.0.0.1",
    "PROXY_PORT": "8080",
    "PROXY_AUTHENTICATION": "FALSE"
}
```
If you use a PAC file to configure your proxy, the PAC file location is
C:/Config. Configure your proxy.json as follows:

```json
{
  "ENABLE_FLAG": "TRUE",
  "USE_PROXY_PAC": "TRUE",
  "PAC_URL": "c:/config/example.pac",
  "USE_PROXY_STATIC": "FALSE",
  "PROXY_HOST": "",
  "PROXY_PORT": "",
  "PROXY_AUTHENTICATION": "",
  "USER_NAME": "",
  "PASSWORD": ""
}
```

You must configure the proxy correctly for it to take effect. Data access issues occur if the format is correct, but the proxy server is inaccessible.

3. Restart the Intelligence Server.

4. Verify your data.

If you verify data from a URL, you can import data from a source that was previously inaccessible.

You should always restart the Intelligence Server after changing the proxy configuration or the PAC file.

To verify that the proxy is taking effect, retrieve a proxy log by enabling the RESTWrapper Info log in the MicroStrategy Diagnostics and Performance Logging Tool.
How to Configure a Proxy in Workstation (Windows or Mac) and the Intelligence Server (Linux)

The steps for other platforms are similar to configuring the Intelligence Server for Windows. The difference is the `proxy.json` file location and the method to restart the server.

**On Windows**, `proxy.json` is located in `C:\Program Files\MicroStrategy\Workstation`.

**On Linux**, `proxy.json` is located in `<InstallPath>/install directory`. `<InstallPath>` is chosen by the user.

**On MacOS**, `proxy.json` is located in `/Applications/MicroStrategy Workstation.app/Contents/Frameworks/OneTierDashboardwindow.framework/Resources` by default.

**Configuration Field Descriptions**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENABLE_FLAG</td>
<td>Enable or disable the proxy. If the flag is false, the proxy does not take effect, regardless of the values in other fields.</td>
</tr>
<tr>
<td>USE_PROXY_PAC</td>
<td>Set USE_PROXY_PAC or USE_PROXY_STATIC to TRUE to turn on the PAC mode or static mode.</td>
</tr>
<tr>
<td>PAC_URL</td>
<td>Specify the path of the PAC file. Both local and online files are supported, such as <code>Z:/BIN/X64/demo.pac</code> or <code>http://test/demo.pac</code>.</td>
</tr>
<tr>
<td>USE_PROXY_STATIC</td>
<td>If this flag is set to TRUE, PROXY_HOST and PROXY_PORT are set as the static proxy.</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>server.</td>
</tr>
<tr>
<td>PROXY_HOST</td>
<td>The static proxy host name or IP address</td>
</tr>
<tr>
<td>PROXY_PORT</td>
<td>The static proxy host port (numbers only)</td>
</tr>
<tr>
<td>PROXY_AUTHENTICATION</td>
<td>Static proxy authentication flag. If this flag is true, use specified USER_NAME and PASSWORD to access the static proxy server. This is supported only in static mode, PAC mode is currently not supported.</td>
</tr>
<tr>
<td>USER_NAME</td>
<td>User name to access the static proxy server</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Password to access the static proxy server</td>
</tr>
</tbody>
</table>

The **ENABLE_FLAG** is set to **FALSE** by default. This means any proxy configured in this file takes effect when you configure the system proxy. If you want to use proxy, turn on the proxy feature by setting **ENABLE_FLAG** to **TRUE**.

To verify that the proxy is taking effect, retrieve a proxy log by enabling the RESTWrapper Info log in the MicroStrategy Diagnostics and Performance Logging Tool.

**Priority**

There are two levels of proxy settings in MicroStrategy.

- The higher priority level is the server-level configuration, which is configured in `proxy.json`.

- The lower priority level is the system configuration, which is an environment variable in Linux or MacOS, and internet properties in Windows.
Limitation

The proxy setting is global. MicroStrategy uses the same proxy, even if a different user logs in.
DEPLOYING MICROSTRATEGY WEB AND MOBILE SERVER
This chapter describes the procedure to deploy a project to your user community using MicroStrategy Web and Mobile Server. The process of deploying the .NET version of MicroStrategy Web or MicroStrategy Mobile Server on Windows with Microsoft Internet Information Services (IIS) is explained in detail.

Steps to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP) in a Linux and Windows environment with various Web and application servers are also explained in detail. MicroStrategy Web (JSP) and Mobile Server (JSP) are platform-independent and can be deployed using different combinations of operating systems, Web servers, and application servers.

Web application servers are not MicroStrategy products, so detailed steps cannot be provided for every combination of application server and operating system. This chapter supplies instructions for a few of the most common combinations. The procedures for different operating systems are similar, but you should refer to the vendor-provided information and documentation for details specific to your configuration, or contact MicroStrategy Technical Support.

MicroStrategy Web simplifies the job of deploying to large user groups because end users' machines only need a supported browser. MicroStrategy Web can be accessed from any supported browser because no code must be downloaded. Working as a thin client, MicroStrategy Web provides the functionality that end users and power users require to take full advantage of the MicroStrategy product suite.

Deploying with IIS (Windows)

Microsoft IIS can be used to deploy MicroStrategy Web and MicroStrategy Mobile Server:
Deploying MicroStrategy Web

The ASP.NET version of MicroStrategy Web can be deployed with IIS only on Windows.

For information on supporting IIS 7.x, see and Supporting IIS 7.0.x or IIS 7.5.x as a web server for MicroStrategy Web or Mobile Server, page 83.

- You must have administrative privileges to deploy MicroStrategy Web for your project. If this is the first time you are logging in and you have not changed the default MicroStrategy administrative login, you can use Administrator as the login with no password. After the first time, the user name and password should be changed for security purposes.

- The Microsoft Windows' Users group must have read and execute permissions to all of the files within the MicroStrategy common files folder. This ensures that Internet Information Services has the required permissions to host MicroStrategy Web. By default, this folder is stored in the following directory location:
  - 64-bit Windows environments:
    - C:\Program Files (x86)\Common Files\MicroStrategy
    - C:\Program Files (x86)\MicroStrategy

To connect MicroStrategy Web to your Intelligence Server

1. On the Windows Start menu, point to Programs, then to MicroStrategy Tools, and then choose Web Administrator. The MicroStrategy Web Administrator page opens. This is the page where you connect MicroStrategy Web to the Intelligence Server.

2. Type the name of your Intelligence Server in the Add a server manually box on the MicroStrategy Web Administrator page.
3. Click Connect. All projects loaded on the Intelligence Server are now available from MicroStrategy Web. Click the Home icon to see the list of projects loaded on the Intelligence Server you specified.

4. Send your users the URL:

   http://webservername/MicroStrategy/asp/

   where webservername is the name of the computer hosting your Web server. For example, if the name of your Web server machine is Web_Srv1, then the URL your users would use to access MicroStrategy Web would be

   http://Web_Srv1/MicroStrategy/asp

You have manually connected MicroStrategy Web to the Intelligence Server.

You can also connect automatically whenever MicroStrategy Web Server or Intelligence Server starts.

   To make MicroStrategy Web connect to the Intelligence Server automatically


   2. Select the Automatically connect to Intelligence Server when Web Server or Intelligence Server is restarted option and click Save.

Deploying Mobile Server

The ASP.NET version of MicroStrategy Mobile Server can only be deployed with IIS only on Windows.

   • For information on supporting IIS 7.x, see Supporting IIS 7.0.x or IIS 7.5.x as a web server for MicroStrategy Web or Mobile Server,
• You must have administrative privileges to deploy MicroStrategy Mobile Server for your project. If this is the first time you are logging in and you have not changed the default MicroStrategy administrative login, you can use Administrator as the login with no password. After the first time, the user name and password should be changed for security purposes.

• The Users group for Microsoft Windows must have read and execute permissions to all of the files within the MicroStrategy common files folder. This ensures that IIS has the required permissions to host MicroStrategy Mobile Server. By default, this folder is stored in the following directory location:
  - 64-bit Windows environments: C:\Program Files\Common Files\MicroStrategy
  - 64-bit Windows environments: C:\Program Files (x86)\Common Files\MicroStrategy

To connect MicroStrategy Mobile Server to your Intelligence Server

1. In Windows, go to Start > Programs > MicroStrategy Tools > Mobile Administrator.

2. Type the name of your Intelligence Server in the Add a server manually box on the MicroStrategy Mobile Server Administrator page.

3. Click Connect.

4. Click Mobile Configuration to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this
You have manually connected MicroStrategy Mobile Server to the Intelligence Server.

You can also connect automatically whenever MicroStrategy Mobile Server or Intelligence Server starts.

To make MicroStrategy Mobile Server connect to the Intelligence Server automatically


2. Select the **Automatically connect to Intelligence Server when Mobile Server or Intelligence Server is restarted** option.

3. Click Save.

General Steps to Deploy MicroStrategy JSP Applications

After you have installed MicroStrategy Web (JSP), Mobile Server (JSP) you can deploy and configure it for your specific environment. The configuration and deployment steps are provided in the Web server and application server sections in this chapter. The table below lists the general steps for all environments.

**MicroStrategy Web JSP**

1. Log on to the application server by using the proper user name and password.

3. To increase the performance of the application before proceeding with the deployment, see the *Performance-based setup information* section, if available, for your environment and configure as necessary. Also, after deploying MicroStrategy Web (JSP) on your machine, there may be a few performance-based setup steps that you should complete.

4. Choose the desired deployment method and follow the deployment procedure.

5. Log on to the MicroStrategy Web Administrator Page.


7. Start working with the application.

You must perform extra configuration steps to allow graphs to support non-Western European fonts on MicroStrategy Web (JSP) for a Linux system. For more information, see *Graph and Document Support of Non-Western European Fonts*, page 915 of *Appendix B, Troubleshooting*.

**MicroStrategy Mobile Server (JSP)**

1. Log on to the application server by using the proper user name and password.


3. To increase the performance of the application before proceeding with the deployment, see the *Performance-based setup information* section, if available, for your environment and configure as necessary. Also, after deploying MicroStrategy Mobile Server (JSP) on your machine, there may be a few performance-based setup steps that you should complete.
4. Choose the desired deployment method and follow the deployment procedure.

5. Log on to the MicroStrategy Mobile Server Administrator Page.

6. From the MicroStrategy Mobile Server Administrator Page, configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, as well as steps to deploy and configure a certificate server for Mobile Server, see the MicroStrategy Mobile Administration Guide.

7. Start working with the application.

Locating the WAR File

The MicroStrategy JSP applications are packaged within single files for each application, called a WAR (Web ARchive) file, following J2EE specifications. You must deploy the WAR file to run the JSP applications in your application server environment.

The WAR files are placed in the folder you specified when installing MicroStrategy JSP applications. The default locations are as follows:

<table>
<thead>
<tr>
<th>Component and WAR File Name</th>
<th>Default WAR File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroStrategy Web (JSP)</td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\WebJSP</td>
</tr>
<tr>
<td>MicroStrategy Mobile Server</td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\WebJSP</td>
</tr>
</tbody>
</table>
|                             | • Linux environments: INSTALL_PATH/WebUniversal  
  Where INSTALL_PATH is the directory you specified as the MicroStrategy install directory during installation. |
<table>
<thead>
<tr>
<th>Component and WAR File Name</th>
<th>Default WAR File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>(JSP) MicroStrategyMobile.war</td>
<td>Files\MicroStrategy\Mobile Server JSP</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Mobile Server JSP</td>
</tr>
<tr>
<td></td>
<td>• Linux environments: INSTALL_PATH/Mobile Server JSP Where INSTALL_PATH is the directory you specified as the MicroStrategy install directory during installation.</td>
</tr>
</tbody>
</table>

To deploy the application, you must follow a set of steps that are specific to the application server you are using. For more details, see the application server vendor documentation or follow the instructions within this guide.

After deploying a WAR file, you can view the WEB-INF folder, which contains a subfolder named log. The log folder retains all the log files. For more information on the directory structure after deploying the WAR file, see *Directory Structure After Deploying the WAR File*, page 466.

**Locating the Configuration Files**

If you have configured your application server to deploy exploded WAR files, the sys_defaults and sys_definitions XML configuration files are located in /WEB-INF/xml in the exploded directory. In Tomcat, these exploded directories are located in the webapps/subdirectory of the Tomcat installation.

- When deploying MicroStrategy Web (JSP), the exploded directory is named MicroStrategy by default.
- When deploying MicroStrategy Mobile Server (JSP), the exploded directory is named MicroStrategyMobile by default.
If you have configured your application server to deploy an unexploded WAR file, the configuration files are created in the system's default temporary file directory. For Windows systems, the temporary file directory is usually `C:\Users\[Username]\AppData\Local\Temp\[Number]\microstrategy\web-[Version]J\`. For example, `C:\Users\Administrator\AppData\Local\Temp\1\microstrategy\web-10.4.0800.0156J\`. For Linux systems, the temporary file directory is usually `/tmp/` or `/var/tmp/`.

- When deploying MicroStrategy Web (JSP), a `/microstrategy/web-Version/` folder is created in the temporary file directory, where `Version` is the version number for the MicroStrategy Web (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.

- When deploying MicroStrategy Mobile Server (JSP), a `/microstrategy/mobile-Version/` folder is created in the temporary file directory, where `Version` is the version number for the MicroStrategy Mobile Server (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.

**Directory Structure After Deploying the WAR File**

The following tables show the default directory structure after deploying MicroStrategy WAR files in your application server.

### MicroStrategy Web

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>\assets</td>
<td>Supporting files</td>
</tr>
</tbody>
</table>
### Installation and Configuration Guide

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>\html</td>
<td>Supporting files</td>
</tr>
<tr>
<td>\images</td>
<td>All image files</td>
</tr>
<tr>
<td>\javascript</td>
<td>Interface JavaScript files</td>
</tr>
<tr>
<td>\jsp</td>
<td>Interface JSP code files</td>
</tr>
<tr>
<td>\META-INF</td>
<td>Configuration files</td>
</tr>
<tr>
<td>\plugins</td>
<td>Plug-in files for customizations</td>
</tr>
<tr>
<td>\resBundles</td>
<td>Flash descriptor files</td>
</tr>
<tr>
<td>\style</td>
<td>Interface style files</td>
</tr>
<tr>
<td>\swf</td>
<td>Supporting files for widgets</td>
</tr>
<tr>
<td>\VisFramework</td>
<td>Supporting files for visualizations</td>
</tr>
<tr>
<td>\WEB-INF</td>
<td>Configuration information for MicroStrategy Web</td>
</tr>
</tbody>
</table>

### MicroStrategy Mobile Server

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>\assets</td>
<td>Supporting files</td>
</tr>
<tr>
<td>\html</td>
<td>Supporting files</td>
</tr>
<tr>
<td>\images</td>
<td>All image files</td>
</tr>
<tr>
<td>\javascript</td>
<td>Interface JavaScript files</td>
</tr>
<tr>
<td>\jsp</td>
<td>Interface JSP code files</td>
</tr>
<tr>
<td>\META-INF</td>
<td>Configuration files</td>
</tr>
<tr>
<td>\plugins</td>
<td>Plug-in files for customizations</td>
</tr>
<tr>
<td>\style</td>
<td>Interface style files</td>
</tr>
</tbody>
</table>
Deploying with WebLogic and Apache (Solaris)

This section provides information used to deploy and configure MicroStrategy JSP applications on the Oracle Solaris operating system, using Apache as the Web server and Oracle WebLogic Server as the application server. It provides information for WebLogic 10.3. You can also the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP).

This section includes the following information:

- **WebLogic Paths and Folder Locations, page 469**: Default folder structure for each version of WebLogic

- **Preconfiguration Information, page 469**: Configuration that must occur before you begin deploying MicroStrategy Web (JSP) and Mobile Server (JSP).

- **Deploying MicroStrategy Web and Mobile Server, page 471**: Instructions for deploying the application

- **Re-deploy the Application, page 481**: Instructions for re-deploying the application

- **Performance-Based Setup Information, page 481**: Optional configuration settings to increase the application’s performance.
The additional configuration steps are not required for MicroStrategy Web (JSP) to run, but these settings can increase its performance. Review the performance-based setup information prior to deploying the system to see if these changes are of interest to you.

WebLogic Paths and Folder Locations

This section presents the default folder structure for each version of WebLogic, and provides the variable used throughout the rest of this chapter to represent the WebLogic mydomain folder path.

Each version of WebLogic is installed with a different default path to the WebLogic mydomain folder. When deploying MicroStrategy Web (JSP), you must make some changes within the WebLogic folders. Thus, it is important to understand the WebLogic folder structure for the version of WebLogic you are using. The following path reflects the default folder structure for WebLogic 12.2.1.3: WEBLOGIC_HOME/user_projects/domains/mydomain/

- WEBLOGIC_HOME is the WebLogic Server home path.

The folder structures are configurable and your organization may have changed the default names or path.

Throughout this chapter, the WebLogic mydomain folder is referred to as WEBLOGIC_MYDOMAIN_FOLDER. This variable refers to the WebLogic mydomain folder in whatever location it resides on your system. The location of this variable is based on the version of WebLogic and whether your organization has changed the version's default name or path.

Preconfiguration Information

This section provides the preconfiguration information necessary to deploy your MicroStrategy JSP applications on your machine.
This section supports the configuration outlined in the following table. While your setup may vary slightly, for example, you may have different versions of these applications, the overall process remains the same.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Oracle Solaris 10.x or 11.x (on SPARC)</td>
</tr>
<tr>
<td>Web server</td>
<td>Apache 2.x</td>
</tr>
<tr>
<td>Application server</td>
<td>WebLogic 10.3</td>
</tr>
</tbody>
</table>
| JDK              | Oracle JDK 1.6.0 or 1.7.0            

You can download the JDK here.

For information on the version numbers supported or certified by MicroStrategy, see the MicroStrategy Readme.

For information on installing these products, see [http://www.oracle.com/technology/products/weblogic/integration/index.html](http://www.oracle.com/technology/products/weblogic/integration/index.html)

Before you start the deployment process, locate the machine name and IP address.

Setting up Apache Web Server to Proxy Requests to WebLogic

You can have the Apache Web server and WebLogic Server running independently on the same machine, but to configure Apache to proxy the desired requests to the WebLogic Server, you must install a plug-in provided by WebLogic. Complete the instructions at the following URLs to install and configure the plug-in.

For WebLogic 10.3, the URL is: [http://e-docs.bea.com/wls/docs100/plugins/apache.html](http://e-docs.bea.com/wls/docs100/plugins/apache.html)
Install the plug-in with the WebLogic installation in the following location:

```
WEBLOGIC_HOME/wlserver_10.3/server/plugin/solaris/sparc/
```

where `WEBLOGIC_HOME` is the path to the WebLogic Server.

To increase the performance of MicroStrategy Web (JSP), you can complete additional setup configurations before the deployment. For more information, see *Performance-Based Setup Information, page 481*.

## Deploying MicroStrategy Web and Mobile Server

When your machine has been configured with the necessary settings, you can deploy the JSP version of MicroStrategy Web and Mobile Server with Apache and WebLogic.

The *Performance-Based Setup Information, page 481* section provides information on additional settings to increase application performance. These additional settings are not required but can increase the performance of MicroStrategy Web (JSP). Review this information prior to deployment to see if these options are of interest to you.

You can deploy MicroStrategy Web and Mobile Server using one of the following deployment methods:

- The automatic deployment feature is the easiest and fastest way. See *Deploying automatically (development mode), page 472*. Choose the development mode in the `WEBLOGIC_MYDOMAIN_FOLDER/bin/startWebLogic.sh` file, within the WebLogic Server folder structure.

- The manual deployment feature can be used for environments where the server is running in production mode and the automatic
deployment is turned OFF. For more information, see Deploying manually (production mode), page 475.

Deploying automatically (development mode)

When automatic deployment is set to ON, as soon as you place a WAR file in the /WEBLOGIC_MYDOMAIN_FOLDER/autodeploy folder, the application is automatically deployed.

With this method you can deploy from:

- A duplicate WAR file. When you deploy from a duplicate WAR file, you are required to manually configure the web.xml file within the WAR file to allow access to certain folders. Once this configuration is complete and the WAR file is recompiled, the JSP application can be deployed using the single WAR file.

- An exploded directory where all the files contained in the WAR file were extracted. When you deploy from an exploded directory, all of the files and folders within the WAR file are exposed to WebLogic. This allows WebLogic access to the required folders so that it can make any necessary configuration changes to files in the exploded directory.

To automatically deploy MicroStrategy JSP applications from a duplicate WAR file

1. Locate the WAR file for your MicroStrategy JSP application, as described in Locating the WAR File, page 464.

2. [Optional] Rename the WAR file to a name you can easily identify and remember. This name is the context_name used in the uniform resource locator (URL) to access the file.
If you do not change the name of the file, remember to replace context_name with MicroStrategy when accessing the application from the URL.

To modify the web.xml file for multiple MicroStrategy deployments

If you are deploying more than one MicroStrategy environments on the same WebLogic application server, prior to deployment, you must modify the web.xml file as described below.

1. Unzip the WAR file by using the following command:
   
   ```
   jar -xvf FileName.war
   ```
   
   Where FileName is the name of the WAR file for your MicroStrategy JSP application.

2. Open the web.xml file located in the /WEB-INF directory.

3. Modify the contextPath parameter. By default, this parameter does not have a value. Type a unique string for the value of the contextPath parameter. For example, type WebDep2.


5. Zip the WAR file by using the following command:
   
   ```
   jar -cvf FileName.war *
   ```
   
   Where FileName is the name of the WAR file for your MicroStrategy JSP application

To deploy the WAR file

1. Transfer the WAR file to the following directory:

   ```/WEBLOGIC_MYDOMAIN_FOLDER/autodeploy```

2. The application is automatically deployed. To add and connect to an Intelligence Server, see Configuring administrative access to MicroStrategy JSP applications, page 478.
To increase the performance of MicroStrategy Web JSP, you can configure additional settings after deployment. For more information, see *Performance-Based Setup Information, page 481*.

To automatically deploy MicroStrategy JSP applications from an exploded directory

The WAR file must be uncompressed by the same user who started the application.

1. Locate the WAR file for your MicroStrategy JSP application, as described in *Locating the WAR File, page 464*.

2. Create the following new folder:

   `/home/username/context_folder`

   where `username` is your account name used to access the Web server machine, and `context_folder` is the name of the new folder.

   You can create the new folder anywhere **except** in the following location:

   `/WEBLOGIC_MYDOMAIN_FOLDER/autodeploy`

3. Copy the WAR file to the new folder.

4. To explode the WAR file inside the folder you created, run the following command:

   `# jar -xvf FileName.war`

   Where `FileName` is the name of the WAR file for your MicroStrategy JSP application.

5. Delete the WAR file by using the following command:
Déploiement manuellement (mode production)

Avec un déploiement manuel, vous pouvez déployer des applications JSP MicroStrategy à partir de:

- Un fichier WAR duplicat. Lorsque vous déployez à partir d’un fichier WAR duplicat, vous êtes requis de configurer manuellement le fichier web.xml dans le fichier WAR pour autoriser l’accès à certaines dossiers. Une fois cette configuration terminée et le fichier WAR recopié, l’application JSP peut être déployée à l’aide du fichier WAR seul.

- Un dossier décomprimé où tous les fichiers contenus dans le fichier WAR ont été extraits. Lorsque vous déployez à partir d’un dossier décomprimé, tous les fichiers et dossiers contenus dans le fichier WAR sont exposés à WebLogic. Cela permet à WebLogic d’accéder aux dossiers requis pour effectuer toute configuration nécessaire à des fichiers dans le dossier décomprimé.
Perform the deployment in the `/WEBLOGIC_MYDOMAIN_FOLDER/autodeploy` directory.

To manually deploy MicroStrategy JSP applications from a duplicate WAR file

1. Locate the WAR file for your MicroStrategy JSP application, as described in *Locating the WAR File, page 464*.

   If you are deploying more than one MicroStrategy environment on the same WebLogic application server, prior to deployment, you must modify the `web.xml` file.

2. Unzip the WAR file by using the following command:

   ```
   #jar -xvf FileName.war
   
   Where `FileName` is the name of the WAR file for your MicroStrategy JSP application.
   ```

3. Open the `web.xml` file located in the `/WEB-INF` directory.

4. Modify the `contextPath` parameter. By default, this parameter does not have a value. Type a unique string for the value of the `contextPath` parameter. For example, type `WebDep2`.

5. Save the `web.xml` file.

6. Zip the WAR file by using the following command:

   ```
   #jar -cvf FileName.war *
   
   Where `FileName` is the name of the WAR file for your MicroStrategy JSP application.
   ```

7. Transfer the WAR file to the `/WEBLOGIC_MYDOMAIN_FOLDER/autodeploy` directory.

8. Open the WebLogic Server Administration Console (WLS Admin Console) by typing the following address:

   ```
   http://IP address:port/console/
   ```
where IP address is the IP address of the machine on which you installed the WebLogic application server and port is the port number for the WebLogic application server.

9. Type a valid user ID and password at the prompt. The user ID and password are the ones you specified when installing the WebLogic Server on your machine.

10. To complete this operation, see Configuring from the WebLogic Server Administration Console, page 478.

To manually deploy MicroStrategy JSP applications from the exploded directory

1. Locate the WAR file for your MicroStrategy JSP application.

2. Create a folder in the WEBLOGIC_MYDOMAIN_FOLDER/autodeploy directory and transfer the WAR file to this directory.

3. Unzip the WAR file using the following command:

   ```bash
   #jar -xvf FileName.war
   ```

   Where FileName is the name of the WAR file for your MicroStrategy JSP application.

4. Open the WebLogic Server Administration Console by accessing the following address:

   ```http://IP address:Port/console/```

   where IP address is the IP address of the machine on which you installed the WebLogic application server and Port is the port number for the WebLogic application server.

5. Type a valid user ID and password at the prompt. The user ID and password are the ones you specified when installing the WebLogic Server on your machine.
Configuring from the WebLogic Server Administration Console

To configure from the WebLogic Server Administration Console, refer to your WebLogic Server Administration Console documentation on steps to install a web application.

Once you have installed the JSP version of MicroStrategy Web and Mobile Server as a WebLogic Server Administration Console web application, you have completed the steps required to deploy the application.

To launch the administrative page for MicroStrategy Web (JSP), Mobile Server (JSP) (JSP), see Configuring administrative access to MicroStrategy JSP applications, page 478.

To increase the performance of MicroStrategy Web (JSP), you can configure additional settings after deployment. For more information, see Performance-Based Setup Information, page 481.

Configuring administrative access to MicroStrategy JSP applications

Before you start MicroStrategy Web (JSP), Mobile Server (JSP) (JSP), you must configure their administrator pages.

To configure access to the MicroStrategy JSP applications

1. The following table lists the URL to access MicroStrategy Web Administrator and MicroStrategy Mobile Server Administrator, for each deployment method.

   The servlet names are case-sensitive. Make sure to use the correct case when typing the mstrWebAdmin name.

<table>
<thead>
<tr>
<th>Deployment Method</th>
<th>URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebLogic Server</td>
<td>/mstrWebAdmin</td>
<td>Access MicroStrategy Web Administrator</td>
</tr>
<tr>
<td>Mobile Server</td>
<td>/mstrWebAdmin</td>
<td>Access MicroStrategy Mobile Server Administrator</td>
</tr>
</tbody>
</table>

© 2019, MicroStrategy Inc.
If the application server is enabled with security, a dialog box related to the administrator authentication opens.

<table>
<thead>
<tr>
<th>Deployment Method</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Automatic deployment</strong></td>
<td>Access the Administrator page from a web browser using this URL:</td>
</tr>
<tr>
<td></td>
<td>• For Web (JSP): <a href="http://IPaddress:7001/context_folder_web/servlet/mstrWebAdmin">http://IPaddress:7001/context_folder_web/servlet/mstrWebAdmin</a></td>
</tr>
<tr>
<td></td>
<td>In the URL listed above, <code>context_folder_web</code> is the name of the folder where the Web (JSP) application was exploded and <code>IPaddress</code> is the IP address of your machine.</td>
</tr>
<tr>
<td></td>
<td>• For Mobile Server (JSP): <a href="http://IPaddress:7001/context_folder_mobile/servlet/mstrWebAdmin">http://IPaddress:7001/context_folder_mobile/servlet/mstrWebAdmin</a></td>
</tr>
<tr>
<td></td>
<td>In the URL listed above, <code>context_folder_mobile</code> is the name of the folder where the Mobile Server application was exploded and <code>IPaddress</code> is the IP address of your machine.</td>
</tr>
<tr>
<td><strong>Manual deployment</strong></td>
<td>Access the Administrator page from a browser using this address:</td>
</tr>
<tr>
<td></td>
<td>• For Web (JSP): <a href="http://IPaddress:7001/Web_name/servlet/mstrWebAdmin">http://IPaddress:7001/Web_name/servlet/mstrWebAdmin</a></td>
</tr>
<tr>
<td></td>
<td>In the URL listed above, <code>IPaddress</code> is the IP address of your machine. Replace the <code>Web_name</code> variable with the name you specified in the deployed name field when configuring Web (JSP) from WebLogic Server Administration Console, for example, <code>MyWebApp</code>.</td>
</tr>
<tr>
<td></td>
<td>• For Mobile Server (JSP): <a href="http://IPaddress:7001/Mobile_name/servlet/mstrWebAdmin">http://IPaddress:7001/Mobile_name/servlet/mstrWebAdmin</a></td>
</tr>
<tr>
<td></td>
<td>In the URL listed above, <code>IPaddress</code> is the IP address of your machine. Replace the <code>Mobile_name</code> variable with the name you specified in the deployed name field when configuring Mobile Server from WebLogic Server Administration Console, for example, <code>MyMobileApp</code>.</td>
</tr>
</tbody>
</table>

2. Type the same user ID and password that was used to start the WebLogic Server on your machine.
In WebLogic, the deployment of a MicroStrategy JSP application automatically associates the WebLogic administrative user with the MicroStrategy JSP application administrator. The WebLogic administrative user is the user who has permissions to start the WebLogic Server on a given machine.

3. If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.

4. If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click **Mobile Configuration** to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the **MicroStrategy Mobile Administration Guide**. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

**Launching the project**

The address to launch MicroStrategy Web (JSP) is different for each deployment method. The table below lists the URL you can use to access MicroStrategy Web (JSP).

<table>
<thead>
<tr>
<th>Deployment Method</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic deployment</td>
<td>Access MicroStrategy Web (JSP) from a web browser using this URL: <a href="http://IPaddress:7001/context_folder/servlet/mstrWeb">http://IPaddress:7001/context_folder/servlet/mstrWeb</a> where context_folder is the name of the folder where the application was exploded and IPaddress is the IP address of your machine.</td>
</tr>
</tbody>
</table>
Re-deploy the Application

If you have already deployed MicroStrategy Web (JSP) with WebLogic and you change any parameters in the web.xml file, you must re-deploy the application using the WebLogic Server Administration Console. This allows the changes to take effect in the deployed application. To re-deploy MicroStrategy Web (JSP), refer to your WebLogic Server Administration Console documentation on steps to re-deploy (update) a web application.

Performance-Based Setup Information

The performance of MicroStrategy Web (JSP) can be increased by configuring it on various component levels. These additional setup settings are not required, but if you want to increase the performance of MicroStrategy Web (JSP), some changes must be done before or after the deployment procedure. This section provides the following configurations:

Setting the Java Heap Size

The Java heap size for the WebLogic Server can be increased by modifying the MEM_ARGS variable in the startWebLogic.sh script:
1. Open the `startWebLogic.sh` script from `/WEBLOGIC_MYDOMAIN_FOLDER/bin/startWebLogic.sh`.

2. Define the following line in the script:
   ```
   MEM_ARGS="-Xms512m -Xmx1024m"
   ```
   This line reflects an initial Java heap size of 512 MB. MicroStrategy recommends the initial java heap size be set at a minimum of 512 MB, assuming the machine has enough memory space. This value may need to be modified to reflect the requirements of your specific environment. Refer to your third-party application server documentation for information on how to determine a satisfactory Java heap size for your environment.

3. Stop and start the application server.

Precompiling JSP Files

To avoid the time taken to load the Web pages in the application server when you access it for the first time, you must precompile the Java Server Pages (JSP) files before deploying the application. Do this by setting the application server to load all the pages in the application before deployment. Thus, when you connect for the first time, the pages are already loaded and the performance is better.

1. Open the `weblogic.xml` file located in the `/WEB-INF` directory.

2. In the `jsp-descriptor` section, set the `keepgenerated` and the `precompile` parameters to `TRUE`, as follows:

   ```xml
   <jsp-descriptor>
     :
     :
     <jsp-param>
       <param-name>keepgenerated</param-name>
       <param-value>TRUE</param-value>
     </jsp-param>
     <jsp-param>
   </jsp-descriptor>
   ```
3. Save the file.

Disable/Relax Auto-Reload Parameters

To disable/relax auto-reload parameters, complete the following steps.

Set the pageCheckSeconds Parameter

The pageCheckSeconds parameter sets the interval, in seconds, at which the WebLogic Server checks to see if JSP files have changed and need recompiling. Dependencies are also checked and recursively reloaded if changed.

You can set the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pages are checked on every request.</td>
</tr>
<tr>
<td>-1</td>
<td>The page is not checked until the server is restarted. Any classes used by the JSP page that live in the servlet classpath are also reloaded.</td>
</tr>
<tr>
<td>n</td>
<td>Interval (in seconds) in which WebLogic Server checks if JSP files have changed. For example, if this is set to 1, WebLogic checks the pages every second to see if the JSP has changed and needs recompiling.</td>
</tr>
</tbody>
</table>
To Set the pageCheckSeconds Parameter

1. Open the weblogic.xml file located in the /WEB-INF directory.

2. In the jsp-descriptor section, set the pageCheckSeconds parameter value. For example, the following code sets the value to -1:

```xml
<jsp-descriptor>
    ...
    <jsp-param>
        <param-name>pageCheckSeconds</param-name>
        <param-value>-1</param-value>
    </jsp-param>
    ...
</jsp-descriptor>
```

3. Save the file.

Set the WebLogic Reload Period Parameter

In WebLogic, the Reload Period parameter sets how often WebLogic checks whether a servlet has been modified. If the servlet has been modified, WebLogic reloads it. As the MicroStrategy Web (JSP) servlets do not change after they have been deployed, MicroStrategy recommends that you disable the reload period by setting it to -1. A value of -1 means never reload, and a value of 0 means always reload.

Use the appropriate procedure below, depending on whether you have MicroStrategy Web (JSP) deployed as a duplicate WAR file.
To Set the WebLogic Reload Period

1. Open the `weblogic.xml` file located in the `/WEB-INF` directory.

2. In the `container-descriptor` section, set the `servlet-reload-check-secs` parameter value. For example, the following code sets the value to -1:

   ```xml
   <container-descriptor>
   <servlet-reload-check-secs>-1</servlet-reload-check-secs>
   </container-descriptor>
   ```

3. Save the file.

Configuring Apache Web Server to Serve Static Files

Because Web servers are tuned to effectively serve static files, the perceived performance of MicroStrategy Web (JSP) is significantly enhanced if image, style sheet, and JavaScript files are served via the Apache Web server, and the WebLogic Server handles only the servlet requests. Do this by editing two main parameters, `Alias` and `MatchExpression`, in the Apache configuration file `httpd.conf`.

- The `Alias` parameter is used to create a virtual directory in the Apache Web server. The virtual directory is needed to serve static files such as images, style sheets, and JavaScript.

- The `MatchExpression` parameter is used to configure the Apache plug-in so that the WebLogic Server handles only the servlet requests.
To Configure the Apache Web Server to Serve Static Files

1. To change the **Alias** parameter, add the following lines in the `httpd.conf` file:

   ```
   Alias /MicroStrategy/images/
   WEBLOGIC_MYDOMAIN_FOLDER/autodeploy/MicroStrategy/images/
   <Directory WEBLOGIC_MYDOMAIN_FOLDER/autodeploy/MicroStrategy/images>
   Options Indexes MultiViews
   AllowOverride None
   Order allow, deny
   Allow from all
   </Directory>
   ```

   These code excerpts assume the application name is **MicroStrategy**. See *Deploying with WebLogic and Apache (Solaris), page 468* for information on default folder structure.

2. Repeat the previous step for the JavaScript and style sheet folders, replacing the word `images` in the previous code with the folder name where the JavaScript and style sheet files are located.

3. Change the **MatchExpression parameter** by typing

   ```
   */servlet/*
   ```

   in the **MatchExpression parameter**. For example,

   ```
   <IfModule mod_weblogic.c>
   WebLogicHost 10.15.133.56
   WebLogicPort 7001
   MatchExpression */servlet/*
   </IfModule>
   ```

4. Stop and start the Apache Web server using the commands `apachectl start` and `apachectl stop`. 
The Web server now serves image (GIF), style sheet (CSS), JavaScript, and all other static files, thus reducing the load on the application server and increasing the application's performance.

Deploying with WebSphere and IBM HTTP Server (AIX)

This section provides information used to deploy and configure MicroStrategy JSP applications on an AIX machine using the WebSphere Server and the IBM HTTP Web Server. You can use the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP).

This section includes the following information:

Deploying MicroStrategy Web and Mobile Server

Once your machine has the necessary settings configured, you can deploy MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) on the WebSphere machine.

The Performance-Based Setup Information, page 496 section provides information on additional settings to increase application performance. These additional settings are not required, but can increase the performance of MicroStrategy Web (JSP).

Launching the WebSphere Administrative Console

The WebSphere Administrative Console can be accessed only if the WebSphere server is started on the machine.
To start the Websphere Application Server

1. Execute the `startServer` script as follows:
   
   ```
   cd WAS_HOME/bin
   #./startServer.sh SERVER_NAME
   ```

   Typically, `server1` is the default server name in WebSphere.

2. Ensure that the Administrative Server has started successfully.
   Execute the following commands:
   
   ```
   cd WAS_HOME/bin
   #./serverStatus.sh -all
   ```

To launch the WebSphere Administrative Console

In a browser, type the URL for the administrative console. The URL is of the following form:

```
http://IP Address:Port.ibm/console
```

where `IP Address` is the IP address of the computer on which you installed the WebSphere application server and `Port` is the port number for the WebSphere Administrative Console. Refer to your third-party WebSphere documentation to confirm the default port number for the administrative console.

Starting the WebSphere default application server

After you launch the WebSphere Administrative Console, you can deploy MicroStrategy Web (JSP) by starting the default application server.

This is applicable for WebSphere Network Deployment Edition or WebSphere Enterprise Edition. For WebSphere Express or WebSphere
Base Editions, there is no distinction between an administrative server and a default server. The `StartServer.sh` command starts the default application server automatically.

To start the default application server

When the WebSphere Administrative Console opens, a tree view is displayed.

1. Expand the **Servers** node, or click the link to expand the view.
2. Click the **Applications Servers** link.
3. Select the box next to the application server to start.
4. Click **Start**.

Installing the Enterprise Application

To install the Enterprise Application

1. Expand **Applications**, and then **Enterprise Applications** to display a list of installed applications.
2. Click **Install**.

Preparing for the application installation

The following steps describe the settings that must be specified for the installation.

Copy the WAR file (see *Locating the WAR File, page 464*) for the MicroStrategy JSP placement to the `WAS_HOME/installableApps` directory, where `WAS_HOME` is the WebSphere application server home path.
To specify settings for the installation

1. To begin the installation for IBM WebSphere, go to **Applications > Application Types > WebSphere enterprise applications**.

2. Click **Install**.

3. You must specify the path to the WAR file by selecting either the local file system or remote file system option. For local file systems, you can click browse to navigate to the location of the WAR file. For remote file systems, type in the full path for the location of the WAR file.

4. Click **Next**.

5. Select to perform either a **Fast Path** or **Detailed** installation. Either type of installation can support the deployment of MicroStrategy JSP applications.

6. Select the **Generate Default Binding** check box, and ensure that the **Override existing bindings** check box is cleared.

7. Click **Next**.

8. Select the **Precompile Java Server Pages files** check box.

9. Specify the value for the **Directory to Install Application** as `${APP_INSTALL_ROOT}/DefaultNode`.

10. Specify an **Application Name** of your choice.

11. Ensure that the **Override class reloading settings for Web and EJB modules** check box is cleared.

12. Click **Next**.

13. Select the **Web Tier** check box and click **Next**.

14. Type a suitable name for **ContextRoot**, which is case-
sensitive. Do not include .war in the name for ContextRoot as this can cause errors when attempting to start the application.

The URLs to access MicroStrategy JSP applications contain ContextRoot, which should be replaced by the name of your choice. For example, MicroStrategy Web JSP uses the URL http://machine-name/ContextRoot/servlet/mstrWeb and you can use the default name of the WAR file, which is MicroStrategy.

15. Click **Next**.

16. Review the summary and click **Finish**.

17. To grant access to these resources, map the admin role to the users or groups that will be given the administrator privileges for your MicroStrategy JSP application. To access these options in WebSphere, expand the Security options, and then click **Global Security**.

   Security must be enabled for the WebSphere Server for this feature to work.

Regenerating plugin-cfg.xml

1. Expand **Environment**, and then click **Update global WebServer Plug-in configuration**.

2. Click **OK**, and then click **Save to master configuration**.

Restarting the application server

This section explains how to stop and start the application server. Performing these steps stops and starts all the applications running on the application server. To stop and start only the application in which you are working, see *To start the Web module, page 492*. 
To restart the application server

The option to stop and start the application server through the administrative console is available only for the Websphere Network Deployment and Websphere Enterprise Editions. To stop and start the application server in Websphere Express and Websphere Base editions, see below.

1. Expand **Servers**, and then click the **WebSphere Application Servers** link.

2. Select the box next to the application server you want to stop, and click **Stop**.

3. Select the application server you want to start and click **Start**.

To stop and start the application server in Websphere Express and Websphere Base editions, use the following commands:

- `stopServer.sh server1` to stop the application server
- `startServer.sh server1` to start the application server.

Starting a single JSP application

This process starts only a single JSP application, rather than all the applications running on the application server. To stop and start all applications, see *Restarting the application server*, above.

To start the Web module

1. Expand **Applications**, then expand **Application Types**, and then select **WebSphere enterprise applications**.

2. Select the box next to the application to start and click **Start**.
Configuring administrative access to MicroStrategy JSP applications

The administrative pages for your MicroStrategy JSP applications are accessible only to users with an admin role. To create the set of users and passwords that are authorized for this access, you must create the necessary role mapping between these users and the admin role for the MicroStrategy JSP application. The steps to perform this setup are given above in the section Preparing for the application installation, page 489. For more information, refer to your IBM documentation.

To configure administrative access to MicroStrategy JSP applications

1. Access the servlet by typing the following URL in a web browser:

   - For Web (JSP):  
     http://IPAddress/ContextRootWeb/servlet/mstrWebAdmin

     In the URL listed above, ContextRootWeb is the name you provided for the ContextRoot for Web Module box in the section Preparing for the application installation, page 489. For example, the default name of the WAR file, which is MicroStrategy.

   - For Mobile Server (JSP):  
     http://IPAddress/ContextRootMobile/servlet/mstrWebAdmin

     In the URL listed above, ContextRootMobile is the name you provided for the ContextRoot for Web Module box in the
For example, the default name of the WAR file, which is MicroStrategyMobile.

The servlet names are case-sensitive. Use the correct case when typing the mstrWebAdmin name. If the application server is enabled with security, a dialog box related to the administrator authentication opens.

2. Type the user ID and password assigned with the admin role.

3. After you are authenticated, add and connect to an Intelligence Server.

4. If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click Mobile Configuration to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the MicroStrategy Mobile Administration Guide. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

5. If you are deploying MicroStrategy Web (JSP), proceed to launch the project.

Launching the project

After configuring the MicroStrategy Web Administrator page, you must follow the steps described in this section to launch your project.
To launch the project

1. Start the Apache Web server by using the following command:

```
# /usr/HTTPServer/bin/apachectl start
```

For example, if the IBM HTTP server is installed in the default location /usr/IBMIHS, then use the following command:

```
/usr/IBMIHS/bin/apachectl
```

2. In a Web browser, specify the following URL:

```
http://MachineName/ContextRoot/servlet/mstrWeb
```

Alternatively, you can use the IP address of your machine for remote access, as shown below:

```
http://IPAddress/ContextRoot/servlet/mstrWeb
```

In these addresses, ContextRoot is the name you provided for the context root for Web Module box on Preparing for the application installation page. For example, the default name of the WAR file, which is MicroStrategy. For more information, refer to Preparing for the application installation, page 489.

You can now access the MicroStrategy Web (JSP) application.

Uninstalling MicroStrategy JSP applications

You can uninstall the MicroStrategy JSP applications through the WebSphere Administrative console.

1. Go to Applications > Application Types > WebSphere enterprise applications.

2. Select the desired MicroStrategy JSP application.

3. Click Uninstall.

4. Save the configuration in the master repository.
Performance-Based Setup Information

The performance of MicroStrategy Web (JSP) can be increased by configuring it on various component levels. These additional settings are not necessary, but can increase the performance of MicroStrategy Web (JSP).

Setting the Java Heap Size

You can increase the Java heap size for a given application server by configuring the WebSphere Administrative Console:

1. Access the Administrative Console.
2. Expand the Servers node.
3. Click the Application Servers link to view the list of application servers.
4. Click the application server name, scroll to Additional Properties and click Process Definition.
5. Click JVM Settings to set the Java heap size settings.
   MicroStrategy recommends that you initially set the Java heap size to a minimum of 500 MB, assuming the machine has enough memory space.
   This value may need to be modified to reflect the requirements of your specific environment. Refer to your third-party application server documentation for information on how to determine a satisfactory Java heap size for your environment.
6. Click Apply.
7. Stop and start the application server.
Precompiling JSP Files

To avoid the time taken to load the Web pages in the application server when you access it for the first time, you must precompile the Java Server Pages (JSP) files. Precompilation can be done during deployment by selecting the **Enable pre-compile of JSPs** setting. Otherwise, it can be done after deploying the application.

To precompile the JSPs after deployment, set the application server to load all the pages in the application. Then when you connect for the first time, the pages are already loaded and performance is improved.

Before you precompile the JSP files, make sure that:

- The MicroStrategy Web (JSP) application is deployed in the WebSphere environment.
- You know the defined application name and the Web Module’s name. You can retrieve these names from the Administrative Console. Locate the application name under the Enterprise Applications node. Locate the Web Module name by expanding the application and clicking **Web Modules**. The default name is Web Tier.

---

**To precompile the JSP files**

1. Change the directory to `WAS_ROOT/bin`.

2. Run the following command:

   ```bash
   ./JspBatchCompiler.sh -enterpriseapp.name ApplicationName -webmodule.name webModule -cell.name cellName -node.name nodeName -server.name serverName -keepgenerated TRUE
   ```
If the administrative server is running in a security enabled mode, you are prompted for the user ID and password to connect to the Admin server.

```
username userID
password password
```

For each JSP file compiled without error, the following message appears: Code generation successful.

**Configuring the IBM HTTP Server to Serve Static Files**

The IBM HTTP Server (Web server) is tuned to effectively serve static files. As a result, perceived performance is greatly enhanced if you configure the IBM HTTP Server to serve image, style sheet, and JavaScript files. This also reduces the load on the WebSphere Server so that it can handle only dynamic files while IBM HTTP Server handles static files. This requires that you do the following:

- Configure the application server level to serve the Java Server Pages (JSPs) and servlets, which are dynamic files and handled by WebSphere.

- Configure the Web server level to serve the images, JavaScripts, and style sheets, which are static files and handled by the IBM HTTP Server.

For more information, see the *IBM WebSphere Application Server* redbook covering System Management and Configuration. This book discusses separating static content from dynamic content.

**Deploying with Oracle Glassfish Server (Solaris)**

This section provides information used to deploy and configure MicroStrategy JSP applications on an Oracle Glassfish Server 3.1.x in
a Linux environment. You can use the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP).

- **Locating the WAR File, page 464.**
- **Deploying MicroStrategy Web and Mobile Server, page 499:** Instructions for deploying MicroStrategy Web (JSP), Mobile Server (JSP).
- **Undeploying MicroStrategy JSP Applications, page 509:** Instructions for undeploying MicroStrategy JSP applications.

### Deploying MicroStrategy Web and Mobile Server

Once your machine has been configured, you can deploy MicroStrategy Web (JSP), Mobile Server (JSP) with Oracle Glassfish Server 3.1.x.

The administration and deployment tools in Oracle Glassfish Server have the same interface regardless of the operating system on which they run. Therefore, the deployment process is the same for all operating systems, and is described below. There are some minor changes in the Windows environment, which are highlighted where necessary.

### Launching the Oracle Glassfish Server Administration Console

This procedure describes the steps to launch the Oracle Glassfish Server.

*The Oracle Glassfish Server is installed. This installation should also include a default domain, commonly named `domain1`. If you plan to use a different domain, refer to your third-party Oracle documentation for creating a domain.*
Copy the WAR file (see *Locating the WAR File, page 464*) for your MicroStrategy JSP application to the same machine as the Oracle Glassfish Server, or to a location that is accessible to the Oracle Glassfish Server machine.

To launch the Oracle Glassfish Server Administration Console

1. Navigate to the following directory in the command prompt:  
   `InstallDir/bin`
   
   where `InstallDir` is the directory where you installed Oracle Glassfish Server.

2. Type the following command to start the domain:
   ```
   asadmin start-domain --domaindir
   DomainDirectory DomainName
   ```
   
   where:
   - `DomainDirectory` is the path you defined when creating the domain. You can remove the `--domaindir` option if the domain uses the default directory.
   - `DomainName` is the name of the domain you created in the previous steps.

   For example, to start domain1, which is the default domain, type the following command:
   ```
   asadmin start-domain domain1
   ```

3. Access the Oracle Glassfish Server Administration Console by typing the following URL:
   ```
   http://MachineName:PortNumber
   ```
   
   where:
- **MachineName** is the IP address or the name of the machine where you installed Oracle Glassfish Server.

- **PortNumber** is the port number you provided when creating the domain. The default port number is 4848.

4. If prompted, type the user name and password that you provided when creating the domain.

**Deploying your MicroStrategy JSP application**

After launching the Oracle Glassfish Server Administration Console, follow the steps below to deploy MicroStrategy JSP applications as a WAR file.

**Save the WAR file (see Locating the WAR File, page 464)** to the same machine as the Oracle Glassfish Server, or to a location that is accessible to the Oracle Glassfish Server machine.

Access to the administrative pages for MicroStrategy Web (JSP) and Mobile Server (JSP) can be granted by using the `admin` security role and the associated `mstradmin` group. Granting this access to users can be done within the Oracle Glassfish Server Administration Console after deploying the WAR file. While this default behavior supports most deployment requirements, if you have specific security requirements for your system, you must modify the security role details prior to deploying the WAR file, as described in Deploying MicroStrategy Web and Mobile Server, page 499.

---

**To deploy MicroStrategy JSP applications as a WAR file**

1. Access the Administration Console by typing the following URL:

   `http://MachineName:PortNumber`

   where:
• MachineName is the IP address or the name of the machine where you installed Oracle Glassfish Server.

• PortNumber is the port number you provided when creating the domain. The default port number is 4848.

2. If prompted, type the user name and password that you used to create the domain.

3. Expand the Tree pane on the left side of the Administration Console.

4. Click Applications.

5. Click Deploy.

6. Select Local Packaged File or Directory That Is Accessible from GlassFish Server, and then click Browse Files.

   Selecting the WAR file in this manner is recommended as the Packaged File to Be Uploaded to the Server option uploads the WAR file via HTTP, which can require considerable time and system resources.

7. Browse to the location where you saved the MicroStrategy JSP application WAR file.

8. Once you select the appropriate WAR file, click Choose File.

9. From the Type drop-down list, select Web Application.

10. In the Context Root field, type the context root for the application, which is included in various URLs for the application:

    • The URL to access MicroStrategy Web (JSP) (http://
IPAddress:PortNumber
/ContextRoot/servlet/mstrWeb) includes the applications context root, which should be replaced by any name of your choice. For example, you can use the default name of the WAR file, which is MicroStrategy.

- The URL to access the MicroStrategy Mobile Server Administrator Page
  
  http://IPAddress:PortNumber
  /ContextRoot/servlet/mstrWebAdmin) includes the applications context root, which should be replaced by any name of your choice. For example, you can use the default name of the WAR file, which is MicroStrategyMobile.

11. In the **Application Name** field, type a descriptive name to distinguish the application from within the Administration Console.

12. In the **Virtual Servers** list, select the appropriate server.

13. Select or clear the additional deployment option check boxes according to your requirements.

   It is recommended you select the Precompile JSPs check box to quickly load the Web pages in the application server when you access it for the first time.

14. Click **OK**.

**Configuring administrative access to MicroStrategy JSP applications**

For security purposes, you must only assign certain users the administrative authorization to access the MicroStrategy Web
Administrator, Mobile Server Administrator. To do this, users need to be assigned to the mstradmin group, which is part of the admin security role.

Oracle Glassfish Server supports the following authentication realms out-of-the-box:

- File realm
- Administration realm
- Certificate realm

A realm, also called a security policy domain or security domain, is a scope over which a common security policy is defined and enforced by the security administrator of the security service. For more information, see the following resource:

http://docs.oracle.com/cd/E18930_01/html/821-2435/ggkuk.html#gkbiy

In Oracle Glassfish Server, the file realm is the default realm. For controlling access to the Administration pages, you can create users and user groups and assign the mstradmin group to users in your security realm.

To create users that are assigned to the mstradmin group in the file realm

1. In the Administration Console, from the Tree pane on the left, click server (Admin Server). Ensure that the server is running or click Start to start the server.

2. From the Tree pane on the left, go to Configuration > server-config > Security > Realms and select file..

3. Click Manage Users.

4. Click New.
5. Type the following information for the new user:

- **User ID**: The ID that the user provides when authenticating with the system.

- **Group List**: The groups that the user is a member of. Type `mstradmin` to provide the user administrative access to MicroStrategy Web Administrator and MicroStrategy Mobile Server Administrator.

- **New Password**: The password used to authenticate a user.

- **Confirm New Password**: A confirmation of the password, required when creating a new user.

6. Click **OK**.

7. In the Administration Console, from the Tree pane on the left, select **server (Admin Server)**.

8. Click **Restart**.

Managing the admin security role for specialized group authentication requirements

MicroStrategy provides a descriptor file, `glassfish-web.xml`, which enables Oracle Glassfish Server to map the existing users or groups to security roles. This file is located within the MicroStrategy JSP application WAR files, and after deployment can be found in the `WebApplicationRootDir/WEB-INF` folder.

By default, the **admin security role** and its **mstradmin group** defined in this `glassfish-web.xml` file can be used to grant administrative access to the MicroStrategy Web Administrator and Mobile Server Administrator. This provides administrative access without having to make any modifications to `glassfish-web.xml`. In these scenarios, you can use the steps provided in *Deploying your MicroStrategy JSP application*, page 501 and *Configuring administrative access to*...
MicroStrategy JSP applications, page 503 to complete the deployment and authentication requirements.

While this default behavior supports most deployment requirements, you can modify this glassfish-web.xml file if you have specialized group authentication requirements to use a group other than the default mstradmin group defined for the admin security role. Any groups that are used must be included as part of the admin security role.

⚠ Any changes made to the glassfish-web.xml file must be done prior to deploying the MicroStrategy JSP application.

The contents of this file are as follows, which may differ depending on your installation of Oracle Glassfish Server:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Servlet 3.0//EN" "http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">
<glassfish-web-app>
    <security-role-mapping>
        <role-name>admin</role-name>
        <group-name>mstradmin</group-name>
    </security-role-mapping>
</glassfish-web-app>
```

Once you make any changes to this file, you must deploy the application (see Deploying your MicroStrategy JSP application, page 501) and assign the security role to the necessary user accounts (see Configuring administrative access to MicroStrategy JSP applications, page 503).
Accessing the MicroStrategy JSP application administrative page

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

To access the MicroStrategy JSP application administrative page

1. Access the servlet by typing the following URL in a Web browser:

   - For Web (JSP):
     
     http://IPAddress:
     PortNumber/ContextRootWeb/servlet/mstrWebAdmin

     In the URL listed above, ContextRootWeb is the name you provided for the ContextRoot for Web Module box in the section Deploying your MicroStrategy JSP application, page 501. For example, you can use the default name of the WAR file, which is MicroStrategy. The default port number is 8080.

   - For Mobile Server (JSP):
     
     http://IPAddress:
     PortNumber/ContextRootMobile/servlet/mstrWebAdmin

     In the URL listed above, ContextRootMobile is the name you provided for the ContextRoot for Web Module box in the section Deploying your MicroStrategy JSP application, page 501. For example, you can use the default name of the WAR
file, which is MicroStrategyMobile. The default port number is 8080.

The servlet names are case-sensitive. Use the correct case when typing the mstrWebAdmin name. If the application server is enabled with security, a dialog box related to the administrator authentication opens.

2. Type the user ID and password for a user who is a member of the mstradmin group, as described in Configuring administrative access to MicroStrategy JSP applications, page 503.

3. After you are authenticated:
   - If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.
   - If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click Mobile Configuration to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the MicroStrategy Mobile Administration Guide. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

4. If you are deploying MicroStrategy Web (JSP), proceed to launch the MicroStrategy project. For more information, see Connecting to the Web (JSP) project page, page 508 immediately below.

Connecting to the Web (JSP) project page

After restarting the application server, follow the steps described here to connect to the project page.
To connect to the Web (JSP) project page

In a Web browser, type the following URL:

http://MachineName:PortNumber/ContextRoot

If you have used all the default values, you can access the following URLs:

http://localhost:8080/MicroStrategy/

or

http://localhost:8080/MicroStrategy/servlet/mstrWeb

Undeploying MicroStrategy JSP Applications

Oracle recommends undeploying an application before deploying a newer version. The steps below show you how to undeploy an existing MicroStrategy JSP application, using the Oracle Glassfish Server Administration Console.

To undeploy MicroStrategy JSP applications

1. In the Administration Console, from the Tree pane on the left, click Applications. The Applications page is displayed.

2. Select the check box for the MicroStrategy JSP application.

3. Click Undeploy.

4. After the undeployment is finished, stop and restart the application server for the changes to take effect.

Deploying with Tomcat (Windows)

This section provides information used to deploy and configure MicroStrategy JSP applications in a Tomcat-only environment. For information on how to configure Tomcat to work with IIS, see
iishowto.html (Tomcat 6.0) in the Tomcat documentation. You can use the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

- **Preconfiguration Information, page 510**: Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP).


### Preconfiguration Information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on Tomcat on your machine.

### Installing the JDK

If you have not installed the Oracle JDK yet, download the file from the website. Be sure to install the JDK and not the JRE software options.

To configure the JDK, a system variable must point to the folder where you install the JDK. If you install the JDK to a simple folder path such as C:\ setting the system variable is easier and more likely to be correct.

### Configuring the JDK

The third-party products discussed below are manufactured by vendors independent of MicroStrategy, and the steps to configure these products is subject to change. Refer to the appropriate Microsoft documentation for steps to access and modify the environment variables.

1. From the **Start** menu, go to **Computer** > **System properties** > **Advanced system settings** > **Environment Variables** > **System
Variables.

2. Under , click **New** to create a system variable.

3. In the **Variable Name** field, type **JAVA_HOME**.

4. In the **Variable Value** field, type the path of the folder where you installed the JDK and click **OK**.

   For example, if the fully qualified path to your JDK executable is `C:\jdk1.6.0\bin\java.exe`, the value of your **JAVA_HOME** variable is `C:\jdk1.6.0`.

   If you have installed JDK under the **Program Files** folder, type `Progra~1` when specifying the folder name in the **Variable Value** box; otherwise the system does not recognize the folder. For example, type `C:\Progra~1\jdk1.6.0` in the **Variable Value** box.

Configuring Tomcat

This procedure assumes that you have downloaded and installed Tomcat on your machine. You can download Tomcat from the Apache website; depending on the version you want to download, you may need to locate the appropriate file in Apache's Archive area. Instructions for downloading and installing Tomcat are also available on the Apache website.

To configure Tomcat, a system variable must point to the folder where you install Tomcat. Installing Tomcat to a simple folder path such as `C:\Tomcat` makes it easier to define the system variable.

The third-party products discussed below are manufactured by vendors independent of MicroStrategy, and the steps to configure these products is subject to change. Refer to the appropriate Microsoft documentation for steps to access and modify the environment variables.
1. From the **Start** menu, go to **Computer > System properties > Advanced system settings > Environment Variables > System Variables**.

2. Click **New** to create a system variable.

3. In the **Variable Name** field, type `CATALINA_HOME`.

4. In the **Variable Value** field, specify the path of the folder where you installed Tomcat and click **OK**. For example, if you installed Tomcat directly to the C drive, the destination folder is `C:\Tomcat`.

   **Note:** If you installed Tomcat under the `Program Files` folder, type `Program~1` when specifying the folder in the Variable Value box. Otherwise, the system does not recognize the folder. For example, type `C:\Program~1\Tomcat` in the Variable Value box.

**Setting the Java heap size**

The Java heap size for the Tomcat can be modified by defining the `JAVA_OPTS` parameter in the `catalina.bat` file. For example, you can define this parameter as follows:

```
JAVA_OPTS = "-Xms1024m -Xmx2048m"
```

This value may need to be modified to reflect the requirements of your specific environment. Refer to your third-party application server documentation for information on how to determine a satisfactory Java heap size for your environment.

**Deploying MicroStrategy Web and Mobile Server**

Assuming you have made all the necessary configurations described above, you can begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) with Tomcat.
Deploying using Tomcat as a stand-alone Web container

To deploy MicroStrategy JSP applications using Tomcat as a stand-alone Web container

1. Locate the WAR file for your MicroStrategy JSP application, as described in *Locating the WAR File, page 464.*

2. Copy the WAR file to the *Tomcat\webapps* folder.

3. From the **Start** menu, select **Run**.

4. Type `cmd` in the Open drop-down list and click **OK**.

5. Browse to the *Tomcat\bin* folder, where *Tomcat* is the folder in which you installed Tomcat. For example, in the command prompt, type

   ```
   cd C:\Tomcat\bin
   ```

6. Click **Enter**. *C:\Tomcat\bin>* is displayed at the command prompt.

7. Type the required commands to start and stop Tomcat, which depends on your version of Tomcat. For example, for Tomcat 7, type `Tomcat7 start` to start Tomcat and type `Tomcat7 stop` to stop Tomcat. Refer to your third-party Apache documentation for information on the commands to start and stop Tomcat.

   If you installed Tomcat under the *Program Files* folder, type `Progra~1` when you change folders in the command prompt.

   Otherwise, the system does not recognize the folder. For example, type `C:\Progra~1\Tomcat\bin` in the command prompt.

Your MicroStrategy JSP application is deployed automatically, based on the following:
If you have configured Tomcat to deploy an exploded WAR file, which is often the default behavior, a folder is created within the Tomcat\webapps folder:

- When deploying MicroStrategy Web (JSP), the folder is named MicroStrategy by default.
- When deploying MicroStrategy Mobile Server (JSP), the folder is named MicroStrategyMobile by default.

If you have configured Tomcat to deploy an unexploded WAR file, the configuration files are created within the system's default temporary file directory. For Windows systems, the temporary file directory is commonly defined by the TMP environment variable:

- When deploying MicroStrategy Web (JSP), a /microstrategy/web-Version/ folder is created within the temporary file directory, where Version is the version number for the MicroStrategy Web (JSP) product. Within this folder location, various configuration files can be found within the WEB-INF folder and its subfolders.
- When deploying MicroStrategy Mobile Server (JSP), a /microstrategy/mobile-Version/ folder is created within the temporary file directory, where Version is the version number for the MicroStrategy Mobile Server (JSP) product. Within this folder location, various configuration files can be found within the WEB-INF folder and its subfolders.

Configuring administrative access your MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must create the users and assign them the role of admin under the Tomcat user configuration file. The steps to configure this access are below.
To configure administrative access to your MicroStrategy JSP applications

1. In the Tomcat\conf folder, open the tomcat-users.xml file in a program that allows you to edit the file, such as Notepad.

2. Add the following tag and save the file:

   <user name="administrator" password="administrator" roles="admin"/>
   You can specify any value in the user name and password fields.

3. Stop and start Tomcat from the command line.

Accessing the MicroStrategy JSP application administrative page

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

1. Access the servlet by typing the following URL in a Web browser:

   - For Web (JSP):
     
     http://localhost:8080/MicroStrategy/servlet/mstrWebAdmin

   - For Mobile Server (JSP):
     
     http://localhost:8080/MicroStrategyMobile/servlet/mstrWebAdmin
The servlet names at the end of the URL are case-sensitive. Make sure to use the correct case when typing the servlet name. If the application server is enabled with security, a dialog box related to the administrator authentication opens.

If you are using Tomcat integrated with IIS, you do not need to specify the port number in the URL. However, when using Tomcat as a stand-alone Web container, you must specify the port number. The default port for Tomcat is 8080.

2. When prompted for a user name and password, use the same values you specified in the tomcat-users.xml file.

3. If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.

4. If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click Mobile Configuration to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the MicroStrategy Mobile Administration Guide. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

Launching the project

In a Web browser, access MicroStrategy Web (JSP) using this URL:

http://localhost:8080/MicroStrategy/servlet/mstrWeb
Deploying with Tomcat (Linux)

This section provides information on how to deploy and configure MicroStrategy JSP applications with Tomcat in a Linux environment. You can use the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

- **Preconfiguration Information, page 517:** Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP).

- **Deploying MicroStrategy Web and Mobile Server, page 519:** Instructions for deploying, including detailed steps.

Preconfiguration Information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications with Tomcat on your Linux machine.

Installing the JDK

If you have not installed Oracle JDK yet, download the shell file [here](#). Be sure to install the JDK and not the JRE software options.

To configure the JDK, a system variable must point to the folder where you install the JDK. If you install the JDK to a simple folder path such as `C:\` setting the system variable is easier and more likely to be correct.

From the location in which to install the JDK, run the file you downloaded:

```
jdk-Version-linux-i586.bin
```

For example, to install version 1.6.0, type the following:

```
jdk-1_6_0-linux-i586.bin
```
Configuring the JDK

1. Open the /etc/profile file using a program that allows you to edit the file.

2. Add the following line:

   ```bash
   export JAVA_HOME=/PathName/jdkVersion;
   ```

   where `PathName` is the destination folder where you installed the JDK and `Version` is the version, such as 1_6_0, of the JDK.

Installing Tomcat

This procedure assumes that you have downloaded and installed Tomcat in a directory named `Tomcat` on your machine. If you have not installed Tomcat yet, download the zip file from the following links:

- Tomcat 7.0.x
- Tomcat 8.0.x

Contact your System Administrator or visit the Apache website for instructions on downloading and installing Tomcat.

Configuring Tomcat

After you install Tomcat, you must configure Tomcat. The Tomcat configuration includes creating the environment variable `CATALINA_HOME` and defining this environment variable to point to the Tomcat directory.

1. Open the `etc/profile` file in a program that allows you to edit the file.

2. Type the following:

   ```bash
   export CATALINA_HOME = /PathName
   ```
where `PathName` is the directory where you have installed Tomcat.

For example,

```
export CATALINA_HOME = /Tomcat
```

### Setting the Java Heap Size

The Java heap size for the Tomcat can be modified by defining the `JAVA_OPTS` parameter in the `catalina.sh` file. For example, you can define this parameter as follows:

```
JAVA_OPTS = "-Xms1024m -Xmx2048m"
```

This value may need to be modified to reflect the requirements of your specific environment. Refer to your third-party application server documentation for information on how to determine a satisfactory Java heap size for your environment.

### Deploying MicroStrategy Web and Mobile Server

After you have performed the configurations described above, you can begin deploying MicroStrategy JSP applications with Tomcat.

#### Deploying using Tomcat as a standalone Web container

To deploy MicroStrategy JSP applications using Tomcat as a standalone Web container

1. Locate the WAR file for your MicroStrategy JSP application, as described in *Locating the WAR File, page 464*.

2. Copy the WAR file to the `Tomcat/webapps` directory.
To start and stop Tomcat from the command line

Type `# $CATALINA_HOME/bin/startup.sh` and click Enter to start Tomcat, which deploys your MicroStrategy JSP applications automatically, based on the following:

- If you have configured Tomcat to deploy an exploded WAR file, which is often the default behavior, a folder is created within the `Tomcat/webapps` folder:
  - When deploying MicroStrategy Web (JSP), the folder is named `MicroStrategy` by default.
  - When deploying MicroStrategy Mobile Server (JSP), the folder is named `MicroStrategyMobile` by default.

- If you have configured Tomcat to deploy an unexploded WAR file, the configuration files are created within the system's default temporary file directory. For Linux systems, the temporary file directory is usually `/tmp/` or `/var/tmp/`:
  - When deploying MicroStrategy Web (JSP), a `/microstrategy/web-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Web (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.
  - When deploying MicroStrategy Mobile Server (JSP), a `/microstrategy/mobile-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Mobile Server (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.
Configuring administrative access to MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must create the users and assign them the role of admin under the Tomcat user configuration file. The steps to configure this access are below.

1. In the Tomcat/conf directory, open the tomcat-users.xml file using a program that allows you to edit the file.

2. Add the following tags and save the file:

   `<role rolename="admin"/>
   <user username="admin" password="admin" roles="admin"/>

   You can specify any value in the user name and password fields. These are used to log in to the MicroStrategy Web Administrator and Mobile Server Administrator pages. The roles field must be admin.

3. Stop and restart Tomcat.

Now you can access and configure your MicroStrategy JSP application, as described in Accessing the MicroStrategy JSP application administrative page, page 521.

Accessing the MicroStrategy JSP application administrative page

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

1. Access the servlet by typing the following URL in a Web browser:
For Web (JSP):
http://
localhost
:8080/MicroStrategy/servlet/mstrWebAdmin

For Mobile Server (JSP):
http://
localhost
:8080/MicroStrategyMobile/servlet/mstrWebAdmin

The servlet names at the end of the URL are case-sensitive. Make sure to use the correct case when typing the servlet name. If the application server is enabled with security, a dialog box related to the administrator authentication opens.

2. When prompted for a user name and password, use the same values you specified in the `tomcat-users.xml` file.

3. After you are authenticated:

   - If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.

   - If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click **Mobile Configuration** to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the *MicroStrategy Mobile Administration Guide*. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

4. If you are deploying MicroStrategy Web (JSP), proceed to launch the MicroStrategy project. In a Web browser, access
MicroStrategy Web (JSP) using the following URL:

http://localhost:8080/MicroStrategy/servlet/mstrWeb

Deploying with SAP NetWeaver (Windows)

This section provides information used to deploy and configure MicroStrategy JSP applications on a Windows machine using the SAP application server. You can use the procedure below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

For detailed deployment instructions, please see: Deploying MicroStrategy Web and Mobile Server, page 523

Deploying MicroStrategy Web and Mobile Server

Once your machine has the necessary settings configured, you can deploy MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) on the SAP-Windows machine.

Deploying MicroStrategy JSP applications with the SAP NetWeaver Application Server

Follow the steps provided in this section to deploy MicroStrategy JSP applications as a WAR file.

To deploy MicroStrategy JSP applications as a WAR file

1. Locate the WAR file for your MicroStrategy JSP application, as described in Locating the WAR File, page 464.

2. Copy the WAR file to the Windows machine hosting your application server. The location in which you store the file is used later and referred to as path_to_war_file.
3. From the Windows **Start** menu, select **Run**.

4. In the **Open** drop-down list, type `cmd`, and click **OK**.

5. Using the command prompt, browse to the following directory within the SAP Application Server installation directory:

   `/usr/sap/SID/Instance_Number/j2ee/deployment/scripts/`

   The `SID` and `Instance_Number` parameters are defined during installation and configuration of the SAP Application Server. The port number above refers to the P4 port number. The default port number is 50004.

6. Type the following command and press Enter to deploy the WAR file:

   ```
   Deploy.bat
   user_name:password@localhost:port_number path_to_war_file
   ```

   The user name and password must have administrative access. The port number above refers to the P4 port number. The default port number is 50004.

7. Access NetWeaver web admin console using the following URL:

   `http://localhost:PortNumber/nwa`

   The `PortNumber` above refers to the J2EE engine port number. The default port number is 50000.

8. Log in as an administrative user.

9. Go to **Operation Management** > **Systems** > **Start & Stop**.

10. Select **Java EE Applications**.

11. Select the MicroStrategy JSP application just deployed from the list.

12. Go to **Application Details** > **Status** > **Start**.

13. Select **On all instances and Set "Started" as Initial State**.
Configuring administrative access to MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must map users or groups to the admin security role. This security role is defined in the MicroStrategy JSP application deployment, within the web-j2ee-engine.xml file. You can modify this file to map users or groups to this admin security role, or include users in the administrators user group.

Accessing the MicroStrategy JSP applications

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

You must have administrative privileges to access the MicroStrategy Web Administrator or Mobile Server Administrator page. For more information, see Configuring administrative access to MicroStrategy JSP applications, page 525.

To access the MicroStrategy Web Administrator or Mobile Server Administrator page

1. Access the servlet by typing the following URL in a Web browser:

   - For Web (JSP):
     
     http://
     
     MachineName
     :PortNumber/MicroStrategy/servlet/mstrWebAdmin

   - For Mobile Server (JSP):
     
     http://
MachineName
:
PortNumber
/MicroStrategyMobile/servlet/mstrWebAdmin

The servlet names at the end of the URLs listed above are case-sensitive. Use the correct case when typing the servlet name.

The login dialog box opens.

2. Specify a user name and password.

3. After you are authenticated:

   - If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.

   - If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click Mobile Configuration to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the MicroStrategy Mobile Administration Guide. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

4. If you are deploying MicroStrategy Web (JSP), access the MicroStrategy Web Application on SAP Web Server by specifying the following URL in the Web browser:

   http://
   MachineName:PortNumber
   /MicroStrategy/servlet/mstrWeb
Deploying with Oracle 10g (Windows)

This chapter provides information used to deploy and configure MicroStrategy JSP applications with Apache as the Web server and Oracle Application Server 10g R3 as the application server. You can use the procedure below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP): For detailed deployment instructions, please see

For detailed deployment instructions, please see: Deploying MicroStrategy Web and Mobile Server, page 527

Deploying MicroStrategy Web and Mobile Server

After your machine is configured, you can start the deployment of your MicroStrategy JSP application with Oracle Application Server 10g R3.

Deploying using the Oracle Enterprise Manager

You can access Oracle Enterprise Manager from the following URL:

http://MachineName:PortNumber/em

Where MachineName is the machine name or IP address of the Oracle machine, and PortNumber is the port number of Oracle Enterprise Manager.

1. Start the Apache Web Server. From the Start menu, go to OracleAS 10g - DEFAULT_HOME1 > Start ApplicationServerName.MachineName.domain.

2. To verify that the Apache Web Server has started, open Oracle Enterprise Manager, select HTTP Server, and then click Start.

3. Select the OC4J instance where you want to deploy your MicroStrategy JSP application. This procedure assumes you are
using the default instance name home. Click **home**. The OC4J: home page opens.

4. Select the **Applications** tab.

5. Click **Deploy**.

6. In the **Archive** area, select **Archive is present on local host**.

7. Click **Browse** to navigate to and select the WAR file for your MicroStrategy JSP application. For more information on locating the WAR file, see *Locating the WAR File, page 464*.

8. In the **Deployment Plan** area, select **Automatically create a new deployment plan** and click **Next**.

9. Enter the **Application Name** and **Context Root**. This section on deploying MicroStrategy Web (JSP) with Oracle 10g uses **MicroStrategy** as the Application Name and */MicroStrategy* as the Context Root. For Mobile Server (JSP), this section uses **MicroStrategyMobile** as the Application Name and */MicroStrategyMobile* as the Context Root.

10. Click **Next**.

**To map a user to the admin security role**

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must assign users the security role of **admin**. In Oracle 10g, the security users and groups are defined in the Oracle Enterprise Manager.

1. In the **Map Security Roles** task name, click the **Go To Task** (pencil) icon.

2. For the **admin** security role, select the **Map Role** (pencil) icon.

3. Select **Map selected users and groups to this role**.
4. In the **Map Role to Users** area, in the **User** field, type the user name to map to the admin security role and click **Add**.

Repeat this step to add all users for whom you want to grant permission to work in the MicroStrategy Web Administrator and Mobile Server Administrator pages.

5. Click **Continue**, and then click **OK**.

6. Click **Deploy**.

7. Stop and restart the Apache Web Server.

Now you can access and configure your MicroStrategy JSP application, as described in *Accessing the MicroStrategy JSP administrative pages, page 529*.

### Accessing the MicroStrategy JSP administrative pages

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

**To access the MicroStrategy JSP administrative pages**

1. In a Web browser, access the administrative page by specifying the following URL:

   - For Web (JSP):
     
     http://
     
     IPAddress:PortNumber
     
     /MicroStrategy/servlet/mstrWebAdmin

   - For Mobile Server (JSP):
     
     http://
     
     IPAddress:PortNumber
     
     /MicroStrategyMobile/servlet/mstrWebAdmin
Where \textit{IPAddress} is the IP address of the Oracle machine and \textit{PortNumber} is the port number used by the Oracle Application Server. The servlet name at the end of the URLs listed above are case-sensitive, so be sure to use the correct case when typing the servlet name.

2. When prompted for a user name and password, specify the values you used earlier when creating the user mapped to the admin security role (see \textit{Deploying using the Oracle Enterprise Manager, page 527 above}).

3. After you are authenticated:
   - If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.
   - If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click \textit{Mobile Configuration} to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the MicroStrategy Mobile Administration Guide. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

4. If you are deploying MicroStrategy Web (JSP), you can now launch the MicroStrategy project. In a Web browser, access MicroStrategy Web (JSP) using this URL:

\begin{verbatim}
http://IPAddress:PortNumber/MicroStrategy/servlet/mstrWeb
\end{verbatim}
Where IPAddress is the IP address of the Oracle machine and PortNumber is the port number used by the Oracle Application Server.

Deploying with JBoss (Windows)

This chapter provides information used to deploy and configure MicroStrategy JSP applications in a JBoss environment. You can use the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

- *Preconfiguration Information, page 531*: configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP).


Preconfiguration Information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on JBoss on your machine.

Installing the JDK

If you have not installed Oracle JDK yet, download it [here](#). Be sure to install the JDK and not the JRE software options.

To configure the JDK, a system variable must point to the folder where you install the JDK. If you install the JDK to a simple folder path such as C:\ then setting the system variable is easier and more likely to be correct.
Configuring the JDK

1. On your Windows machine, go to Start > Computer > System properties > Advanced system settings > Environment Variables.

   The third-party products discussed below are manufactured by vendors independent of MicroStrategy, and the steps to configure these products is subject to change. Refer to the appropriate Microsoft documentation for steps to access and modify the environment variables.

2. Under System Variables, click New to create a system variable. The New System Variable dialog box opens.

3. In the Variable Name box, type JAVA_HOME.

4. In the Variable Value box, specify the destination folder where you installed the JDK and click OK.

   For example, if the fully qualified path to your JDK executable is C:\jdk1.8.0\bin\java.exe, the value of your JAVA_HOME variable is C:\jdk1.8.0.

   If you have installed JDK under the Program Files folder, type Progra~1 in the destination folder; otherwise the system does not recognize the folder. For example, C:\Progra~1\jdk1.8.0.

Installing JBoss

You can download and install JBoss here.

Keep track of the location in which you install JBoss, as this location is used later (referred to as JBOSS_HOME) to configure JBoss with a MicroStrategy JSP application deployment.
Deploying MicroStrategy Web and Mobile Server

Assuming you have made all the necessary configurations described above, you can begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) with JBoss.

Deploying using JBoss as a stand-alone Web container

To deploy MicroStrategy JSP applications using JBoss as a stand-alone Web container

1. Locate the WAR file for your MicroStrategy JSP application, as described in Locating the WAR File, page 464.

2. Copy the WAR file to the JBOSS_HOME\server\default\deploy directory.

3. To start JBoss, browse to JBOSS_HOME\bin. Then run the following command:
   run.bat -b 0.0.0.0

Your MicroStrategy JSP application is deployed automatically, based on the following:

- If you have configured JBoss to deploy an exploded WAR file, which is often the default behavior, a folder is created within the JBOSS_HOME\server\default\deploy directory:
  - When deploying MicroStrategy Web (JSP), the folder is named MicroStrategy by default.
  - When deploying MicroStrategy Mobile Server (JSP), the folder is named MicroStrategyMobile by default.

- If you have configured JBoss to deploy an unexploded WAR file, the configuration files are created within the system's default temporary file directory. For Windows systems, the temporary file directory is
commonly defined by the \texttt{TMP} environment variable:

- When deploying MicroStrategy Web (JSP), a \\
  \\
  /microstrategy/web-\textit{Version}/ folder is created within the \textit{temporal file directory}, where \textit{Version} is the version number for the MicroStrategy Web (JSP) product. Within this folder location, various configuration files can be found within the \texttt{WEB-INF} folder and its subfolders.

- When deploying MicroStrategy Mobile Server (JSP), a \\
  \\
  /microstrategy/mobile-\textit{Version}/ folder is created within the \textit{temporal file directory}, where \textit{Version} is the version number for the MicroStrategy Mobile Server (JSP) product. Within this folder location, various configuration files can be found within the \texttt{WEB-INF} folder and its subfolders.

**Configuring administrative access to MicroStrategy JSP applications**

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must create the users and assign them the role of \texttt{admin} under the JBoss user configuration files. The steps to configure this access are below.

To configure administrative access to MicroStrategy JSP applications

1. 
   
   
   Browse to the directory \texttt{JBOSS_HOME\server\default\conf}, where \texttt{JBOSS_HOME} is the location in which you installed JBoss.

2. 
   
   
   Create the following two files within this directory:

   - users.properties
   - roles.properties
3. Open the `users.properties` file in a text editor.

4. Include one line for each user to grant administrative access to the Web Administrator and Mobile Server Administrator, using the following syntax:
   
   ```
   user_id=user_password
   ```

   For example, you create UserA and UserB with passwords 1234 and 5678 respectively using the following syntax:

   ```
   UserA=1234
   UserB=5678
   ```

5. Save your changes and close the `users.properties` file.

6. Open the `roles.properties` file in a text editor.

7. Include one line for each user you included in the `users.properties` file and grant them administrative access, using the following syntax:
   
   ```
   user_id=admin
   ```

   For example, you define UserA and UserB to have administrative access using the following syntax:

   ```
   • UserA=admin
   • UserB=admin
   ```

8. Save your changes and close the `roles.properties` file.

9. To start JBoss, browse to `JBOSS_HOME\bin`. Then run the following command:
   
   ```
   run.bat -b 0.0.0.0
   ```

Now you can access and configure your MicroStrategy JSP application, as described in *Accessing the MicroStrategy JSP application administrative page*, page 536.
Accessing the MicroStrategy JSP application administrative page

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

To access the MicroStrategy JSP application administrative page

1. In a Web browser, access the administrative page by specifying the following URL:
   - For Web (JSP):
   - For Mobile Server (JSP):

   The servlet names at the end of the URLs listed above are case-sensitive. Make sure to use the correct case when typing the servlet name. If the application server is enabled with security, a dialog box related to the administrator authentication opens.

2. When prompted for a user name and password, type the user name for the administrator user you created in the roles.properties file and the login information in the users.properties file.

3. After you are authenticated:
   - If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.
If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click **Mobile Configuration** to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the MicroStrategy Mobile Administration Guide. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

4. If you are deploying MicroStrategy Web (JSP), proceed to launch the MicroStrategy Web project page. In a Web browser, access MicroStrategy Web project using this URL:

   http://localhost:8080/MicroStrategy/servlet/mstrWeb

### Administering Your MicroStrategy Web Deployment

You can configure and manage MicroStrategy Web connections to Intelligence Servers in the MicroStrategy Web Administrator page.

### Enabling Users to Install MicroStrategy Office from Web

This information applies to MicroStrategy Office, the add-in for Microsoft Office applications which is no longer actively developed.

It was substituted with a new add-in, MicroStrategy for Office, which supports Office 365 applications. The initial version does not yet have all the functionalities of the previous add-in.

⚠️ For more information, see the MicroStrategy for Office page in the 2019 Update 1 Readme and the MicroStrategy for Office Online Help.
From the MicroStrategy Web Administrator page, you can designate the installation directory path to MicroStrategy Office, and also determine whether a link to Office installation information appears in the MicroStrategy Web interface.

You must install and deploy MicroStrategy Web Services to allow the installation of MicroStrategy Office from MicroStrategy Web. For information about deploying MicroStrategy Web Services, see the MicroStrategy for Office Online Help.

To specify the path to MicroStrategy Office and determine whether users can install MicroStrategy Office from Web

1. In Windows, go to Start > Programs > MicroStrategy Tools > ;Web Administrator.

2. Click Connect.


4. In the Path to MicroStrategy Office Installation field, type the base URL of your MicroStrategy Web Services machine, for example:

   http://server:port/Web_Services_virtual_directory/Office

   MicroStrategy Web automatically attaches /Lang_xx/officeinstall.htm to the end of the URL, where

   Lang_xx refers to the currently defined language in MicroStrategy Web. For example, if the language in MicroStrategy Web is set to English, a completed URL may appear as follows:

   http://localhost/MicroStrategyWS/office/Lang_1033/officeinstall.htm

5. Test the URL path by clicking Go.
6. Click your browser's Back button to return to the Web Administration - MicroStrategy Office settings page.

7. To ensure that an Install MicroStrategy Office link is displayed at the top of users' project selection and login pages in MicroStrategy Web, select the Show link to installation page for all users on the Projects and Login pages check box. When users click the 'Install MicroStrategy Office' link, a page opens with instructions on how to install MicroStrategy Office on their machine.

8. Click Save.

Using Absolute Paths to Share Configuration Files

By default, absolute paths are not used for the configuration files of your MicroStrategy Web Mobile deployments. You can modify the microstrategy.xml file to reference configuration files using absolute paths. By using absolute paths, you can allow the same configuration files to be accessed by multiple systems. For example, absolute paths can be used in a clustered environment in which you want all instances of the web server to access the same MicroStrategy Web or Mobile configuration files.

Below is an example of some relative paths that are included in the microstrategy.xml file by default:

```xml
<parameter name="serverConfigFilesDefaultLocation" value="/WEB-INF/xml/" />
<parameter name="serverLogFilesDefaultLocation" value="/WEB-INF/log/" />
```

You can modify these to use absolute paths, as shown in the examples below:

```xml
<parameter name="serverConfigFilesDefaultLocation" value="ABSOLUTE:/usr/User1/MicroStrategy/xml/" />
<parameter name="serverLogFilesDefaultLocation" value="ABSOLUTE:/usr/User1/MicroStrategy/xml/log/" />
```
Configuring Third-Party Data Sources for Importing Data

You can use MicroStrategy Web to import data from different data sources, such as an Excel file, a table in a database, the results of a Freeform SQL query, or other data sources, into MicroStrategy metadata, with minimum project design requirements.

To import data from the following data sources, configure a secure connection between your third-party data source and MicroStrategy Web:

- Dropbox
- Google Analytics
- Google BigQuery
- Google Drive
- Facebook
- Salesforce.com
- Twitter

The steps below show you how to make these third-party data sources available for import into MicroStrategy Web.

Your third-party data source environment contains the data you plan to integrate in MicroStrategy Web. You also need the proper credentials to perform some of the steps below. For example, you need a Salesforce.com login with developer credentials to perform some of the steps below.

- You have deployed MicroStrategy Web so that it uses secure, encrypted communications. For steps to enable secure communications for your
MicroStrategy Web deployment, refer to the System Administration Guide.

- If you are connecting to Salesforce.com, to ensure proper numeric value integration and formatting when using Data Import, your Salesforce.com reports must use the English locale. If you use a different locale for your Salesforce.com reports, you can still integrate this data into MicroStrategy using Data Import if you connect to Salesforce.com using the MicroStrategy ODBC Driver for Salesforce. For steps to configure this type of a connection to Salesforce.com, see MicroStrategy ODBC Driver for Salesforce, page 776.

To configure a connection between a third-party data source and MicroStrategy Web for Data Import

1. Access the administrative options for your third-party data source. For example, if you are integrating data from Salesforce.com, log in to Salesforce.com by accessing https://login.salesforce.com/.

2. You must configure MicroStrategy Web as a remote access application for the third-party data source. For the steps to define remote access applications in your third-party data source, refer to your third-party documentation.

   When configuring MicroStrategy Web as a remote access application, you must define the callback URL as the URL to access MicroStrategy Web, including the event 3172. Depending on how you deployed MicroStrategy Web, the syntax for this URL can take one of the following forms:

   - For MicroStrategy Web (ASP.NET) deployments:

     https://WebServer
For MicroStrategy Web (JSP) deployments:

https://
WebServer
:
PortNumber
/
WebApplicationName
/servlet/mstrWeb?evt=3172&src=mstrWeb.3172

In the example URLs above:

- **WebServer** is the full domain name of your web server that is hosting MicroStrategy Web. Ensure that you use the full domain name rather than using an IP address, as using an IP address can require re-authentication when making the connection.

- **PortNumber** is the port number of your web server.

- **WebApplicationName** is the name of the MicroStrategy Web application. The default name for the MicroStrategy Web application is MicroStrategy.

3. When you save MicroStrategy Web as a remote access application, your third-party data source provides a Client ID and a Client Secret. Save these two values as they are required later to configure the connection.

4. Restart the web server that hosts MicroStrategy Web. The next time you log in to Web and use Data Import, the data source is now an available option.
5. When connecting to your data source using Data Import in MicroStrategy Web, you must supply the Client ID and Client Secret provided by the data source. Additionally, the callback URL is the MicroStrategy Web URL you used above to configure MicroStrategy Web as a remote access application for your data source. The Client ID, Client Secret, and Callback URL are all defined as OAuth parameters of the connection to your data source using Data Import.

Configuring your MicroStrategy Installation

After completing the steps to deploy MicroStrategy Web and Mobile Server, you can continue your setup and configuration. To help guide the rest of your installation and configuration steps, refer to the section Installation and Configuration Checklists, page 110 in Chapter 1, Planning Your Installation, for installation and configuration checklists.
SETTING UP DOCUMENTS AND HTML DOCUMENTS
This section explains the setup required for Intelligence Server to execute HTML documents and Report Services documents on Linux platforms.

Prerequisites ............................................................... 545
Executing Documents and HTML Documents in Linux .......... 546
Configuring your MicroStrategy Installation ....................... 554

Prerequisites

This section assumes the following:

- You are familiar with MicroStrategy Developer and MicroStrategy Intelligence Server.
- You are familiar with MicroStrategy HTML documents and Report Services documents.
- You have a Report Services product license if you are using Report Services documents. HTML documents do not require a Report Services product license.
- You have installed MicroStrategy Developer on a Windows machine.
- You have installed MicroStrategy Intelligence Server on a Linux machine.

Some of the steps described in this document may require root access permissions.

You must perform extra configuration steps to allow Report Services documents to support non-Western European fonts on a UNIX system. For more information, see Graph and Document Support of Non-Western European Fonts, page 915 of Appendix B, Troubleshooting.
Executing Documents and HTML Documents in Linux

A MicroStrategy Report Services document contains objects representing data coming from one or more reports, as well as positioning and formatting information. Report Services documents help format data from multiple reports in a single display and can be used for presentations. When you create a document, you can specify the data that appears, control the layout, formatting, grouping, and subtotaling of data and specify the position of page breaks. In addition, you can insert pictures and draw borders in the document.

In this section, the term document signifies a Report Services document. For additional information on Report Services documents, refer to the Document Creation Guide.

An HTML document is a container for formatting, displaying, and distributing multiple reports on the same page, or at the same time within a project. You can create dossiers and scorecards to display a group of reports within the MicroStrategy platform.

HTML documents are created using MicroStrategy Developer. Before creating or executing HTML documents, you must specify the HTML document directory using the Project Configuration dialog box in MicroStrategy Developer. The HTML document directory stores HTML templates that are required by the MicroStrategy Intelligence Server for executing HTML documents. You can store the HTML document directory on a Linux platform, but you must share the directory with the Windows platform that includes MicroStrategy Developer. For more information on setting up the HTML document directory on a Linux platform, see Setup for Creating and Executing HTML Documents, page 547.
For additional information on HTML documents, see the *HTML Documents* chapter in the Advanced Reporting Guide.

**Setup for Creating and Executing HTML Documents**

HTML documents can only be created with MicroStrategy Developer on a Windows platform, but they can be stored and executed from a directory within a Linux platform. The directory that stores the HTML documents must be accessible on the computer with Intelligence Server and the Windows computer with Developer.

Using the Project Configuration dialog box in MicroStrategy Developer, you must specify the location of the HTML document directory as an absolute path. This document directory can be on a local machine or on a remote machine. Users require appropriate read and write permissions to access this directory. When MicroStrategy Intelligence Server executes HTML documents, it requires read permission to the HTML document directory to access the HTML files.

For the procedure of setting up an HTML document directory between Windows and Linux computers below, the following assumptions are made:

- You have installed MicroStrategy Developer on a Windows computer and installed MicroStrategy Intelligence Server on a Linux computer.

- MicroStrategy Developer can only be installed on a Windows computer.

- MicroStrategy Developer users have at least read permissions to the HTML document directory for executing existing HTML documents. Write permissions to the directory are required for MicroStrategy Developer users to create new HTML documents.
For the file paths described in the procedure below, `machine-name` is used to represent the name of the machine you store the HTML document directory on. For example, if you store the directory on a machine named UNIX1, `machine-name` should be replaced with UNIX1. This machine must have Samba installed to provide access to the folder on a Windows computer.

You must have root permissions on any Linux computer used to set up the HTML document directory. This includes the computer that stores the HTML document directory as well as any computer that must be setup to access the directory.

To set up the HTML document directory

1. Create a directory to hold the HTML document directory on the desired Linux computer. This procedure assumes that the path of the HTML document directory is `machine-name:/share/htmldocuments`. This is the machine that is referenced as `machine-name` in the steps below. To create this directory, enter the commands below:

   ```
   # cd /
   # mkdir share
   # cd share
   # mkdir htmldocuments
   ```

2. Install Samba software on the Linux computer that you created the HTML documents directory in the step above. With this software, the HTML documents directory is accessible to the Windows computer with MicroStrategy Developer installed. You can get the latest version of Samba at [http://www.samba.org](http://www.samba.org).
Notice that Samba uses the .org extension and not the more common .com extension. Using the .com extension takes you to an incorrect website.

3. Share the directory `machine-name:/share` across the network through NFS. For example, you must share `UNIX1:/share`. Make sure read and write permissions are set for the share. This step allows other Linux computers to access the directory.

4. Create a Samba share, named "share", with read and write permissions that points to the directory `machine-name:/share`. For example, you must share `UNIX1:/share`. This step allows Windows computers to access the directory.

The Samba share is created in the Samba `smb.conf` file. For specific instructions on how to setup a Samba share, refer to the Samba website at [http://www.samba.org](http://www.samba.org).

5. Restart Samba.

6. Mount the HTML document directory on the computer that has the Intelligence Server installed on it. Root privileges are required for this.

On the computer with Intelligence Server, type the command `su` and the root password at the command prompt to log in as a superuser, or log in as `root`. The command prompt changes to the pound sign (#). Perform the commands below:

```
In the commands below, `machine-name` refers to the machine name of the computer where you stored the HTML documents directory and created an NFS and Samba share. This may be a different name than the computer that you are mounting the
```
directory on. The final mount command contains a space between /htdocs/documents and /machine-name.

```
cd /
mkdir machine-name
cd machine-name
mkdir share
cd share
mkdir htdocuments
cd /

mount machine-name:/share/htdocs/documents /machine-
name/share/htdocs/documents
```

7. You can cache the connection to the Linux HTML documents directory from the Windows computer so that you are not prompted for authentication each time the directory is accessed:

   a. From the Windows computer that has MicroStrategy Developer installed, click Start, and select Run. The Run dialog box opens.

   b. Type `\\machine-name\share\htdocs/documents`, and click OK to open the top-level shared HTML documents directory. For example, type `\\UNIX1\share\htdocs/documents`.

   This must be performed every time you restart the computer.

8. Using the Project Configuration dialog box in MicroStrategy Developer, set the HTML document directory as an absolute path by following the steps below:
a. In Developer, right-click the project associated with the HTML documents and select **Project Configuration**. The Project Configuration dialog box opens.

b. Expand **Project definition** and click **Advanced**. The Project Configuration - Advanced options are displayed.

c. In the HTML document directory box, type the absolute path `\machine-name\share\htmldocuments`. For example, type `\UNIX1\share\htmldocuments`.

d. Click **OK** to accept the changes.

9. Create a directory named **xsls** under the HTML document directory and copy the XSL files you require for creating HTML documents to the **xsls** directory, `/machine-name/share/htmldocuments/xsls`. If you stored XSL files in a different directory or did not copy them from their original default directory, you must copy them into the new **xsls** directory. For example, the default HTML document directory for the Tutorial project is Program Files\MicroStrategy\Tutorial Reporting.

10. If you want to insert images into the HTML document, create a directory named **images** under the HTML document directory, and copy the images to the directory `/machine-name/share/htmldocuments/images`.

You are now ready to create and execute your HTML documents. Remember to create your HTML documents in the HTML document directory, otherwise, Intelligence Server cannot execute the HTML documents correctly.

**Setup for Executing Existing HTML Documents**

If you have created HTML documents prior to establishing a connection between the HTML document directory on the Linux machine with MicroStrategy Intelligence Server and the Windows
machine with MicroStrategy Developer, you must make sure that all the files used for an HTML document are copied to the shared HTML document directory. After the connection is established, you should always create the HTML documents in the shared HTML document directory. Once the existing files are copied, you can execute the HTML documents using Intelligence Server.

In the procedure of setting up existing HTML documents, the following assumptions are made:

- You have completed all the steps listed in the section Setup for Creating and Executing HTML Documents, page 547.

- The location of the HTML document directory is /machine-name/share/htmldocuments. For the file paths described in the procedure below, machine-name is used to represent the name of the machine on which you store the HTML document directory. For example, if you store the directory on a machine named UNIX1, machine-name should be replaced with UNIX1.

To set up existing HTML documents for execution

1. Copy the HTML file for any existing HTML document to /machine-name/share/htmldocuments.

2. View the source code of each HTML file and copy the XSL file used by each HTML document in an appropriate directory under /machine-name/share/htmldocuments.

   For example, if the location of the XSL file in the source code is xsl="\xs1s\myxsl.xsl, then copy myxsl.xsl to /machine-name/share/htmldocuments/xsls. If the location of the XSL file in the source code is xsl="\myxsl.xsl, then copy myxsl.xsl to /machine-name/share/htmldocuments.

3. View the source code for the images used by each HTML document in an appropriate directory under /machine-
name/share/htmldocuments.

For example, if the location of the image file in the source code is images\myimage.gif, then copy myimage.gif to /machine-name/share/htmldocuments/images. If the location of the XSL file in the source code is \myimage.gif, then copy myimage.gif to /machine-name/share/htmldocuments.

You are now ready to execute your HTML documents.

Setup for Executing Report Services Documents

A MicroStrategy Report Services document is used to format data from multiple reports. These documents can be exported to PDF format. To execute documents and export them to PDF format using MicroStrategy Intelligence Server in a Linux environment, you must perform some additional setup tasks.

When Intelligence Server is running on a Linux platform, all fonts are converted to the Courier New font for:

- Reports exported to PDF format
- Report Services documents
- Graphs contained in HTML documents
- Graphs displayed in MicroStrategy Web

This occurs because the fonts required by the PDF component are missing from Linux machines running Intelligence Server. The missing fonts may include Microsoft True Type fonts.

MicroStrategy does not distribute or license Microsoft fonts, and therefore cannot package Microsoft fonts with Intelligence Server.
To resolve this issue, you must install the font files in the PDFGeneratorFiles folder within the MicroStrategy installation path on the Linux machine, as described below.

To copy fonts to your Linux machine

1. Log in to your Linux machine that hosts Intelligence Server.

2. Install the Microsoft True Type fonts. Refer to the following resources for information on licensing requirements for and installing Microsoft True Type fonts:
   - [http://www.ascendercorp.com/msfonts/msfonts_main.html](http://www.ascendercorp.com/msfonts/msfonts_main.html)

3. Copy the font files into the `INSTALL_PATH/PDFGeneratorFiles` directory, where `INSTALL_PATH` is the directory you specified as the MicroStrategy install directory during installation.

4. To update the list of fonts available, you must restart the Intelligence Server.

Configuring your MicroStrategy Installation

After completing the steps to set up documents and HTML documents for Linux, you can continue your setup and configuration. To help guide the rest of your installation and configuration steps, refer to the section *Installation and Configuration Checklists, page 110* in *Chapter 1, Planning Your Installation*, for installation and configuration checklists.
AUTOMATED INSTALLATION ON WINDOWS
This section explains the various possibilities for performing fully automated and unattended installations within the MicroStrategy platform. This includes customizations to the installation routines available with the product. It explains the different resources you can use to deploy MicroStrategy products through various scenarios including:

- Deploying the MicroStrategy platform across the network through the Microsoft System Management Server (SMS) or its equivalent (for example, IBM Tivoli)
- Embedding the MicroStrategy platform within third party custom applications and other installation routines
- Customizing the MicroStrategy installation to meet the various environment-specific requirements for a given site

This section provides the following information:

- **Installation Log File** .......................................................... 556
- **Methods of Installation** ...................................................... 558
- **Configuring your MicroStrategy Installation** ......................... 622

Automated and silent installations require advanced techniques such as creating and running response.ini files. Therefore, automated and silent installations should be handled by system administrators with full knowledge of the environment and the desired MicroStrategy installation.

Before installing MicroStrategy products, you should refer to *Chapter 1, Planning Your Installation* for important pre-installation information.

### Installation Log File

Before you begin to learn about automated installation options, it is important to know about the installation log file. The setup program
generates a log file in text format. This log file contains records of all actions performed by the setup program and by other executable files related to installation. The installation log file can be particularly helpful if you encounter errors during the installation process. For example, the log can tell you if a registry key or path was not added or if a critical file was not registered successfully. The setup.exe file writes to this log file.

The log file data includes:

- Update dates
- Machine specifications
- User selections
- List of files to be registered
- List of files that do not require registration
- List of registry entries
- Identification of files that fail during registration
- Installation activity such as performance counter loading and DSN creation
- Reboot time file registration results

The default location for the install.log file is:

- **32-bit Windows environments**: Program Files\Common Files\MicroStrategy
- **64-bit Windows environments**: Program Files (x86)\Common Files\MicroStrategy

Both the location and the name can be changed. You can specify the log file name and location in the following places:
- Command line, reading the parameter LogFile. For example:
  
  ```
  setup.exe --LogFile="C:\install.log"
  ```


Methods of Installation

The installation methods discussed in this section are:

Installing and Configuring with a response.ini File

The response.ini file can facilitate the installation and setup of MicroStrategy products by allowing you to progress through the installation and project creation processes with a single keystroke. A response.ini file is an initialization file that is used to send parameters or selections to the MicroStrategy Installation Wizard. This allows you to run it silently as all the options are pre-selected in that file. This section describes how to create and use response.ini for the following tasks:

Configuring a response.ini file to Install MicroStrategy

The response.ini file for installation allows you to automate certain aspects of the installation by modifying a Windows ini-like response file. This option is typically implemented by the following:

- OEM applications that embed MicroStrategy installations within other products
- IT departments who want to have more control over desktop installations

The response.ini file specifies all the selections you want to make during the installation in the MicroStrategy Installation Wizard. You
can either run it with all the MicroStrategy Installation Wizard options that are pre-selected or run it without having to use the wizard at all.

The response.ini file should not be confused with the setup.iss file, which is used by the MicroStrategy Installation Wizard to perform silent installation. When both response.ini and setup.iss are included in the setup, response.ini overrides setup.iss. For details on the setup.iss file, see Silent Installation, page 616.

Component dependencies

When you use a response.ini file to install MicroStrategy products, there are some key dependencies among separate components you should be aware of. The products listed below require either pre-installed software or certain MicroStrategy components to be selected to successfully install the products with a response.ini file:

- MicroStrategy Enterprise Manager requires MicroStrategy Command Manager to be included in the installation.


- MicroStrategy Narrowcast Server Administrator requires SequeLink ODBC Socket Server to be included in the installation.

- MicroStrategy Narrowcast Server Subscription Portal requires SequeLink ODBC Socket Server to be included in the installation.

MicroStrategy Analytic Modules requires MicroStrategy Developer or a combination of MicroStrategy Analyst and MicroStrategy Architect to be included in the installation.

To use MicroStrategy Identity Server and MicroStrategy Identity Manager, they are required to be included in the installation.

MicroStrategy Communicator requires MicroStrategy Identity Server, MicroStrategy Identity Manager, and Platform Analytics to be included in the installation.

Platform Analytics requires MicroStrategy Intelligence Server, MicroStrategy Telemetry, and Command Manager to be included in the installation.

Creating a response.ini file

You can create a response.ini file in any text editor and save the file as response.ini in the desired folder.

⚠️ You must save the file as ANSI encoding.

The following tables describe the parameters and options for the all the sections, such as Installer, Paths, and so on in the response.ini file. It is followed by sample response.ini files for your reference.

ℹ️ The options are case-sensitive, therefore they must be entered as indicated in the tables below.

### Installer

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Installer]</td>
<td>Section that begins the installation.</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ExpressMode = TRUE or False</td>
<td>Indicates whether the installation uses Express Install or Custom Install.</td>
</tr>
<tr>
<td>PropertiesFilesOve rwrite= TRUE or FALSE</td>
<td>Set to TRUE to create new properties files. These files are related to the Tutorial Delivery component. The default is FALSE, which uses the current version of the properties files. The default is FALSE.</td>
</tr>
<tr>
<td>EnableTracing = TRUE or FALSE</td>
<td>Set this option to trace the setup process in a log file that is saved in the Temp folder. The log file records errors that are encountered during the installation. The default is FALSE.</td>
</tr>
<tr>
<td>HideAllDialogs = TRUE or FALSE</td>
<td>Indicates whether the installation uses all default values. FALSE displays all the dialog boxes and you must browse using the Next buttons. The default is FALSE.</td>
</tr>
<tr>
<td>ForceReboot = TRUE or FALSE</td>
<td>TRUE reboots the machine after the installation is done. The default is FALSE.</td>
</tr>
<tr>
<td>PreventReboot = TRUE or FALSE</td>
<td>TRUE prevents the machine from rebooting after installation is done. Note the following conditions:</td>
</tr>
<tr>
<td>CheckTCPIP= TRUE or FALSE</td>
<td>Set to TRUE to check that the TCP/IP network protocol is active. If set to FALSE, the setup doesn't check for it. The default is TRUE.</td>
</tr>
<tr>
<td>CheckIIS= TRUE or FALSE</td>
<td>Set to TRUE to check for Internet Information Services. The default is TRUE.</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateShortcuts =</td>
<td><strong>TRUE or FALSE.</strong> <strong>TRUE</strong> creates the shortcuts for MicroStrategy products, tools, and documentation. The default is <strong>TRUE.</strong></td>
</tr>
<tr>
<td>CheckRenameOperations =</td>
<td><strong>TRUE or FALSE.</strong> In some instances, as a result of a previous installation or an uninstall, certain files may be missing or irreplaceable during installation. Therefore, you are prompted to do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Reboot at the beginning of installation to replace this file. It is recommended that you select this option.</td>
</tr>
<tr>
<td></td>
<td>• Continue with the installation at the risk of the software not functioning properly.</td>
</tr>
<tr>
<td></td>
<td>If you enter <strong>FALSE</strong>, the prompt does not display. The default is <strong>TRUE.</strong></td>
</tr>
<tr>
<td>AnalyticsOverwrite =</td>
<td><strong>TRUE or FALSE.</strong> This option overwrites the Analytics Modules from a previous install. The default is <strong>FALSE.</strong></td>
</tr>
<tr>
<td>TutDeliveryOverwrite =</td>
<td><strong>TRUE or FALSE.</strong> Set this option to overwrite the Delivery Tutorial from a previous installation. The default is <strong>FALSE.</strong></td>
</tr>
<tr>
<td>BackupFiles =</td>
<td><strong>TRUE or FALSE.</strong> If you set the value to <strong>TRUE</strong>, it creates a backup of the following files:</td>
</tr>
<tr>
<td></td>
<td>• *.pds</td>
</tr>
<tr>
<td></td>
<td>• *.xsl</td>
</tr>
<tr>
<td></td>
<td>• *.asp</td>
</tr>
<tr>
<td></td>
<td>• *.css</td>
</tr>
<tr>
<td></td>
<td>• *.js</td>
</tr>
<tr>
<td></td>
<td>• *.sql</td>
</tr>
<tr>
<td></td>
<td>The default is <strong>FALSE.</strong></td>
</tr>
<tr>
<td>RunConfigWizard =</td>
<td><strong>TRUE or FALSE.</strong> When using silent install, set to <strong>FALSE</strong></td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>StopAllServices=</td>
<td>TRUE or FALSE. Set to TRUE to stop all services required to be stopped to complete a MicroStrategy installation. If set to FALSE, the user is prompted if services need to be stopped. The default is TRUE.</td>
</tr>
<tr>
<td>StopIIS =</td>
<td>TRUE or FALSE. Set this option to stop Internet Information Services (IIS) during installation. The default is FALSE.</td>
</tr>
<tr>
<td>EnableASPServices =</td>
<td>TRUE or FALSE. Set this option to enable the ASP MicroStrategy Web Services extensions that IIS Admin requires. The default is FALSE.</td>
</tr>
<tr>
<td>EnableASPNETServices =</td>
<td>TRUE or FALSE. Set this option to enable the ASP.NET MicroStrategy Web Services extensions that IIS Admin requires. The default is FALSE.</td>
</tr>
<tr>
<td>ShowWelcomeScreen =</td>
<td>TRUE or FALSE. Set to TRUE to display the Welcome screen after reboot. The Welcome screen is displayed only once after reboot. The default is TRUE.</td>
</tr>
<tr>
<td>EnterpriseManagerOverwrite =</td>
<td>TRUE or FALSE. If you select FALSE, the warehouse and metadata files are not updated but the rest of the files for Enterprise Manager are updated. This prompt only impacts the files in the Microsoft Access database. The default is FALSE.</td>
</tr>
<tr>
<td>ConfigWizardResponseFile =</td>
<td>Specify the name of the response file for the Configuration Wizard; otherwise, it takes the default name of response.ini. For more details on configuring the response.ini file for the Configuration Wizard, see Configuring your Installation with a response.ini File, page 614.</td>
</tr>
<tr>
<td>LogFile =</td>
<td>Location where the install.log file is generated. If left empty, it takes the default location and file name of:</td>
</tr>
</tbody>
</table>
### 32-bit Windows environments:
C:\Program Files\Common Files\MicroStrategy\install.log.

### 64-bit Windows environments:
C:\Program Files (x86)\Common Files\MicroStrategy\install.log.

## Welcome Dialog

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Welcome]</td>
<td>Section for configuring the Welcome dialog box.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>RemoveAll =</td>
<td>TRUE or FALSE. This option is for the uninstall process only. Setting it to TRUE removes all MicroStrategy products during the uninstall process. The default is FALSE.</td>
</tr>
</tbody>
</table>

For an example of a response file used to uninstall all MicroStrategy products, see *Uninstalling with a response.ini File, page 614*.

## Customer Information Dialog

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[UserRegistration]</td>
<td>Section for specifying the customer information.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>UserFirstName =</td>
<td>Indicates the user name of the currently logged user or a user who is already registered. If no information is provided, you cannot proceed to the next page. If you are installing MicroStrategy Identity Server, this is the first</td>
</tr>
</tbody>
</table>
### License Detail

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[LicenseDetail]</td>
<td>Section that displays license details.</td>
</tr>
<tr>
<td>HideDialog</td>
<td>TRUE or False. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserLastName</td>
<td>Indicates the user name of the currently logged user or a user who is already registered. If no information is provided, you cannot proceed to the next page. If you are installing MicroStrategy Identity Server, this is the last name of the Administrator.</td>
</tr>
<tr>
<td>UserEmail</td>
<td>Indicates the email of the currently logged user, or a user who is already registered. This email address is also used to receive the badge invitation for your MicroStrategy Identity Badge network. If no information is provided, you cannot proceed.</td>
</tr>
<tr>
<td>CompanyName</td>
<td>The name of the company for which the software is registered. The default is the company name in the registry. This is also the default company name when your MicroStrategy Identity Badge network is created.</td>
</tr>
<tr>
<td>LicenseKey</td>
<td>Specify the license key for the software. If you do not specify the license key, the MicroStrategy Installation Wizard will ask for it when it reaches that step. By default, it is blank for a fresh install or displays the license key from a previous install.</td>
</tr>
</tbody>
</table>
### Setup Express Install

<table>
<thead>
<tr>
<th><strong>Options</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>[SetupExpress]</td>
<td>Section that begins the installation.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or False. Indicates whether the installation uses Express Install or Custom Install.</td>
</tr>
</tbody>
</table>

### Choose Destination Location Dialog

<table>
<thead>
<tr>
<th><strong>Options</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>[SuiteTarget]</td>
<td>Section specifying the name of the target directory from where you can run the MicroStrategy products.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>TargetDirectory</td>
<td>Location of the root directory for the Program Files. The default is set to:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environment: C:\Program Files\MicroStrategy</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environment: C:\Program Files (x86)\MicroStrategy</td>
</tr>
</tbody>
</table>

### Paths

<table>
<thead>
<tr>
<th><strong>Options</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>[InitialPaths]</td>
<td>Section for specifying the path for the products that you select to install.</td>
</tr>
<tr>
<td>COMMONFILES =</td>
<td>Location where the common files like response.ini, install.log, and so on will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\Common Files\MicroStrategy</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Developer =</td>
<td>Location where Developer will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Desktop</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files\MicroStrategy\Desktop</td>
</tr>
<tr>
<td>ObjectManager =</td>
<td>Location where Object Manager will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Object Manager</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files\MicroStrategy\Object Manager</td>
</tr>
<tr>
<td>CommandManager =</td>
<td>Location where Command Manager will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Command Manager</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files\MicroStrategy\Command Manager</td>
</tr>
<tr>
<td>EnterpriseManager =</td>
<td>Location where Enterprise Manager will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Enterprise Manager</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files\MicroStrategy\Enterprise Manager</td>
</tr>
<tr>
<td>Server =</td>
<td>Location where the MicroStrategy Intelligence Server will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Web =</td>
<td>Location where MicroStrategy Web will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Web ASPx</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files(x86)\MicroStrategy\Web ASPx</td>
</tr>
<tr>
<td>WebUniversal =</td>
<td>Location where MicroStrategy Web will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Web JSP</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files(x86)\MicroStrategy\Web JSP</td>
</tr>
<tr>
<td>WebServices =</td>
<td>Location where MicroStrategy Web Services will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Web Services</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files(x86)\MicroStrategy\Web Services</td>
</tr>
<tr>
<td>WebServicesUniversal =</td>
<td>Location where MicroStrategy Web Services will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Web Services JSP</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files(x86)\MicroStrategy\Web Services JSP</td>
</tr>
<tr>
<td>Office =</td>
<td>Location where MicroStrategy Office will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Office</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TutorialReporting =</td>
<td><strong>Location where MicroStrategy Tutorial - Reporting will be installed. If left empty, it takes the default location of:</strong></td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Tutorial Reporting</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Tutorial Reporting</td>
</tr>
<tr>
<td>AnalyticsModules =</td>
<td><strong>Location where the Analytics Modules will be installed. If left empty, it takes the default location of:</strong></td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Analytics Modules</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Analytics Modules</td>
</tr>
<tr>
<td>NCSAdminDeliveryEngine =</td>
<td><strong>Location where the Narrowcast Server Delivery Engine will be installed. If left empty, it takes the default location of:</strong></td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Narrowcast Server\Delivery Engine</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Narrowcast Server\Delivery Engine</td>
</tr>
<tr>
<td>SubscriptionPortal =</td>
<td><strong>Location where the Subscription Portal will be installed. If left empty, it takes the default location of:</strong></td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Narrowcast Server\Subscription Portal</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Narrowcast Server\Subscription Portal</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TutorialDelivery</td>
<td>Location where MicroStrategy Tutorial - Delivery will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Narrowcast Server\Tutorial</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Narrowcast Server\Tutorial</td>
</tr>
<tr>
<td>IntegrityManager</td>
<td>Location where MicroStrategy Integrity Manager will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Integrity Manager</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Integrity Manager</td>
</tr>
<tr>
<td>Mobile</td>
<td>Location where MicroStrategy Mobile will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Mobile Clients</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Mobile Clients</td>
</tr>
<tr>
<td>MobileASPPath</td>
<td>Location where MicroStrategy Mobile Server ASP.NET will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Mobile Server ASPx</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Mobile Server ASPx</td>
</tr>
<tr>
<td>MobileJSPPath</td>
<td>Location where MicroStrategy Mobile Server JSP will be installed. If left empty, it takes the default location of:</td>
</tr>
</tbody>
</table>
|                       | • 32-bit Windows environments:
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portlets =</td>
<td>Location where MicroStrategy Portlets will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Portlets</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files\MicroStrategy\Portlets</td>
</tr>
<tr>
<td>MDXCubeProvider =</td>
<td>Location where the MicroStrategy MDX Cube Provider will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\MDX Cube Provider</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files\MicroStrategy\MDX Cube Provider</td>
</tr>
<tr>
<td>GISConnectors =</td>
<td>Location where the MicroStrategy GIS Connectors will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\GISConnectors</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files\MicroStrategy\GISConnectors</td>
</tr>
<tr>
<td>SystemManager =</td>
<td>Location where the MicroStrategy System Manager will be installed. If left empty, it takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\SystemManager</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files\MicroStrategy\SystemManager</td>
</tr>
</tbody>
</table>
Select Components Dialog - Product Visibility State

In the MicroStrategy Installation Wizard, the Select Components dialog box contains checkboxes to select or clear for products to be installed. The [ComponentSelection] options specify whether you want the following products to be visible to the user. In addition, you can set the default selection for each product.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ComponentSelection]</td>
<td>Equivalent to the Select Components dialog box that you see during installation. For the Visible option, you can either enter TRUE to show a product or FALSE to hide it. If you do not specify a TRUE or FALSE value for each product, TRUE is used for all products. For the Select option, you can enter TRUE to select the checkbox next to a product. If you enter FALSE, the checkbox next to the product is not selected.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>AnalyticsModulesVisible =</td>
<td>MicroStrategy Analytics Modules</td>
</tr>
<tr>
<td>ArchitectVisible =</td>
<td>MicroStrategy Architect</td>
</tr>
<tr>
<td>CommandManagerVisible =</td>
<td>MicroStrategy Command Manager</td>
</tr>
<tr>
<td>DeliveryEngineVisible =</td>
<td>MicroStrategy Delivery Engine</td>
</tr>
<tr>
<td>DeveloperVisible =</td>
<td>MicroStrategy Developer</td>
</tr>
<tr>
<td>AnalystVisible =</td>
<td>MicroStrategy Analyst</td>
</tr>
<tr>
<td>EnterpriseManagerVisible =</td>
<td>MicroStrategy Enterprise Manager</td>
</tr>
</tbody>
</table>

© 2019, MicroStrategy Inc.
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FunctionPluginVisible =</td>
<td>MicroStrategy Function Plug-In Wizard</td>
</tr>
<tr>
<td>GISConnectorsVisible =</td>
<td>MicroStrategy GIS Connectors</td>
</tr>
<tr>
<td>IntegrityManagerVisible =</td>
<td>MicroStrategy Integrity Manager</td>
</tr>
<tr>
<td>IServerDistributionServicesVisible =</td>
<td>MicroStrategy Distribution Services</td>
</tr>
<tr>
<td>IServerOLAPServicesVisible =</td>
<td>MicroStrategy OLAP Services</td>
</tr>
<tr>
<td>IServerReportServicesVisible =</td>
<td>MicroStrategy Report Services</td>
</tr>
<tr>
<td>IServerTransactionServicesVisible =</td>
<td>MicroStrategy Transaction Services</td>
</tr>
<tr>
<td>IServerVisible =</td>
<td>MicroStrategy Intelligence Server</td>
</tr>
<tr>
<td>Rvisible =</td>
<td>MicroStrategy Intelligence Server</td>
</tr>
<tr>
<td>MobileServerASPVisible =</td>
<td>MicroStrategy Mobile Server (ASP.NET)</td>
</tr>
<tr>
<td>MobileServerJSPVisible =</td>
<td>MicroStrategy Mobile Server (JSP)</td>
</tr>
<tr>
<td>MobileVisible =</td>
<td>MicroStrategy Mobile</td>
</tr>
<tr>
<td>MobileClientVisible =</td>
<td>MicroStrategy Mobile Client</td>
</tr>
<tr>
<td>NCSAdminVisible =</td>
<td>MicroStrategy Narrowcast Administrator</td>
</tr>
<tr>
<td>ObjectManagerVisible =</td>
<td>MicroStrategy Object Manager</td>
</tr>
<tr>
<td>OfficeVisible =</td>
<td>MicroStrategy Office</td>
</tr>
<tr>
<td>PortletsVisible =</td>
<td>MicroStrategy Portlets</td>
</tr>
<tr>
<td>SequeLinkVisible =</td>
<td>SequeLink ODBC Socket Server</td>
</tr>
<tr>
<td>ServerAdminVisible =</td>
<td>MicroStrategy Server Administrator</td>
</tr>
<tr>
<td>SubscriptionPortalVisible =</td>
<td>MicroStrategy Subscription Portal</td>
</tr>
<tr>
<td>SystemManagerVisible =</td>
<td>MicroStrategy System Manager</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>MDXCubeProviderVisible =</td>
<td>MicroStrategy MDX Cube Provider</td>
</tr>
<tr>
<td>TutorialDeliveryConfigureVisible =</td>
<td>MicroStrategy Tutorial - Delivery Configuration</td>
</tr>
<tr>
<td>TutorialDeliveryInstallVisible =</td>
<td>MicroStrategy Tutorial - Delivery Installation</td>
</tr>
<tr>
<td>WebAnalystVisible =</td>
<td>MicroStrategy Web Analyst</td>
</tr>
<tr>
<td>WebProfessionalVisible =</td>
<td>MicroStrategy Web Professional</td>
</tr>
<tr>
<td>WebReporterVisible =</td>
<td>MicroStrategy Web Reporter</td>
</tr>
<tr>
<td>WebServerASPNETVisible =</td>
<td>MicroStrategy Web Server (ASP.NET)</td>
</tr>
<tr>
<td>WebServerJSPVisible =</td>
<td>MicroStrategy Web Server (JSP)</td>
</tr>
<tr>
<td>WebServicesASPNETVisible =</td>
<td>MicroStrategy Web Services (ASP.NET)</td>
</tr>
<tr>
<td>WebServicesJSPVisible =</td>
<td>MicroStrategy Web Services (JSP)</td>
</tr>
<tr>
<td>UsherServerVisible =</td>
<td>MicroStrategy Identity Server</td>
</tr>
<tr>
<td>UsherNetworkManagerVisible =</td>
<td>MicroStrategy Identity Manager</td>
</tr>
<tr>
<td>UsherProfessionalVisible =</td>
<td>MicroStrategy Communicator</td>
</tr>
<tr>
<td>MessagingServicesVisible =</td>
<td>MicroStrategy Telemetry</td>
</tr>
<tr>
<td>TomcatVisible =</td>
<td>Apache Tomcat</td>
</tr>
<tr>
<td>MySQLVisible =</td>
<td>MySQL</td>
</tr>
<tr>
<td>RVisible =</td>
<td>R Integration Pack</td>
</tr>
<tr>
<td>LibraryWebMobileVisible =</td>
<td>MicroStrategy Library Web and Mobile</td>
</tr>
<tr>
<td>CollaborationServerVisible =</td>
<td>Collaboration Server</td>
</tr>
<tr>
<td>DataServerVisible =</td>
<td>Data Server</td>
</tr>
</tbody>
</table>
Select Components Dialog - Product Selection State

During the installation process in the MicroStrategy Installation Wizard, the Select Components dialog box contains checkboxes to select or clear for products to be installed. You can either specify TRUE to install a product or FALSE to uninstall it. If you do not specify a TRUE or FALSE value for each product, the installation always uses the most recent selection from a previous install.

This means that if you have a product installed and you do not specify a TRUE or FALSE value, the product is upgraded.

If you specify TRUE, the product checkbox is selected. The [ComponentSelection] options specify whether the checkbox for each product will be selected or cleared.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommunityConnectorsVisible</td>
<td>Community Connectors</td>
</tr>
<tr>
<td>PlatformAnalyticsVisible</td>
<td>Platform Analytics</td>
</tr>
<tr>
<td>CertificateManagerVisible</td>
<td>Certificate Store</td>
</tr>
<tr>
<td>AnalyticsModulesSelect</td>
<td>MicroStrategy Analytics Modules</td>
</tr>
<tr>
<td>ArchitectSelect</td>
<td>MicroStrategy Architect</td>
</tr>
<tr>
<td>CommandManagerSelect</td>
<td>MicroStrategy Command Manager</td>
</tr>
<tr>
<td>DeliveryEngineSelect</td>
<td>MicroStrategy Delivery Engine</td>
</tr>
<tr>
<td>DeveloperSelect</td>
<td>MicroStrategy Developer</td>
</tr>
<tr>
<td>AnalystSelect</td>
<td>MicroStrategy Analyst</td>
</tr>
<tr>
<td>EnterpriseManagerSelect</td>
<td>MicroStrategy Enterprise Manager</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>FunctionPluginSelect =</td>
<td>MicroStrategy Function Plug-In Wizard</td>
</tr>
<tr>
<td>GISConnectorsSelect =</td>
<td>MicroStrategy GIS Connectors</td>
</tr>
<tr>
<td>IntegrityManagerSelect =</td>
<td>MicroStrategy Integrity Manager</td>
</tr>
<tr>
<td>IServerDistributionServicesSelect =</td>
<td>MicroStrategy Distribution Services</td>
</tr>
<tr>
<td>IServerOLAPServicesSelect =</td>
<td>MicroStrategy OLAP Services</td>
</tr>
<tr>
<td>IServerReportServicesSelect =</td>
<td>MicroStrategy Report Services</td>
</tr>
<tr>
<td>IServerSelect =</td>
<td>MicroStrategy Intelligence Server</td>
</tr>
<tr>
<td>IServerTransactionServicesSelect =</td>
<td>MicroStrategy Transaction Services</td>
</tr>
<tr>
<td>RSelect =</td>
<td>R, R Integration Pack</td>
</tr>
<tr>
<td>MobileClientSelect =</td>
<td>MicroStrategy Mobile Client</td>
</tr>
<tr>
<td>MobileSelect =</td>
<td>MicroStrategy Mobile</td>
</tr>
<tr>
<td>MobileServerASPSelte =</td>
<td>MicroStrategy Mobile Server (ASP.NET)</td>
</tr>
<tr>
<td>MobileServerJSPSelect =</td>
<td>MicroStrategy Mobile Server (JSP)</td>
</tr>
<tr>
<td>NCSAdminSelect =</td>
<td>MicroStrategy Narrowcast Administrator</td>
</tr>
<tr>
<td>ObjectManagerSelect =</td>
<td>MicroStrategy Object Manager</td>
</tr>
<tr>
<td>OfficeSelect =</td>
<td>MicroStrategy Office</td>
</tr>
<tr>
<td>PortletsSelect =</td>
<td>MicroStrategy Portlets</td>
</tr>
<tr>
<td>SequeLinkSelect =</td>
<td>SequeLink ODBC Socket Server</td>
</tr>
<tr>
<td>ServerAdminSelect =</td>
<td>MicroStrategy Server Administrator</td>
</tr>
<tr>
<td>SubscriptionPortalSelect =</td>
<td>MicroStrategy Subscription Portal</td>
</tr>
<tr>
<td>SystemManagerSelect =</td>
<td>MicroStrategy System Manager</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>MDXCubeProviderSelect =</td>
<td>MicroStrategy MDX Cube Provider</td>
</tr>
<tr>
<td>TutorialDeliveryConfigureSelect =</td>
<td>MicroStrategy Tutorial - Delivery Configuration</td>
</tr>
<tr>
<td>TutorialDeliveryInstallSelect =</td>
<td>MicroStrategy Tutorial - Delivery Installation</td>
</tr>
<tr>
<td>Web Analyst Select =</td>
<td>MicroStrategy Web Analyst</td>
</tr>
<tr>
<td>Web Professional Select =</td>
<td>MicroStrategy Web Professional</td>
</tr>
<tr>
<td>Web Reporter Select =</td>
<td>MicroStrategy Web Reporter</td>
</tr>
<tr>
<td>WebServerASPNETSelect =</td>
<td>MicroStrategy Web Server (ASP.NET)</td>
</tr>
<tr>
<td>WebServerJSPSelect =</td>
<td>MicroStrategy Web Server JSP</td>
</tr>
<tr>
<td>Web Services ASPNETSelect =</td>
<td>MicroStrategy Web Services (ASP.NET)</td>
</tr>
<tr>
<td>Web Services JSPSelect =</td>
<td>MicroStrategy Web Services (JSP)</td>
</tr>
<tr>
<td>Usher Server Select =</td>
<td>MicroStrategy Identity Server</td>
</tr>
<tr>
<td>Usher Network Manager Select =</td>
<td>MicroStrategy Identity Manager</td>
</tr>
<tr>
<td>Usher Professional Select =</td>
<td>MicroStrategy Communicator</td>
</tr>
<tr>
<td>Messaging Services Select =</td>
<td>MicroStrategy Telemetry</td>
</tr>
<tr>
<td>Tomcat Select =</td>
<td>Apache Tomcat</td>
</tr>
<tr>
<td>MySQL Select =</td>
<td>MySQL</td>
</tr>
<tr>
<td>R Select =</td>
<td>R Integration Pack</td>
</tr>
<tr>
<td>Library Web Mobile Select =</td>
<td>MicroStrategy Library Web and Mobile</td>
</tr>
<tr>
<td>Collaboration Server Select =</td>
<td>Collaboration Server</td>
</tr>
<tr>
<td>Data Server Select =</td>
<td>Data Server</td>
</tr>
</tbody>
</table>
## Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommunityConnectorsSelect</td>
<td>Community Connectors</td>
</tr>
<tr>
<td>PlatformAnalyticsSelect</td>
<td>Platform Analytics</td>
</tr>
<tr>
<td>CertificateManagerSelect</td>
<td>Certificate Store</td>
</tr>
</tbody>
</table>

## Installation Files

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[IODSourceLocation]</td>
<td>Section specifying the location of the files required to install the MicroStrategy components you have selected for installation. Specifying the location of the installation files is only required if you have downloaded only a subset of the MicroStrategy installation files and stored some of the files in another location. For steps to determine the files required for your installation, see Creating Custom Installation Packages, page 106.</td>
</tr>
</tbody>
</table>
| Style                        | Determines whether the required installation files are provided in a folder or at a URL. You must define this parameter with one of the following values:  
  - FILESERVER: Type this value if the required installation files are stored in a folder on the local machine or a server machine. You must also provide the location of the files using the SourceLocation parameter.  
  - HTTP: Type this value if the required installation files are stored at an unsecured URL. You must also provide the location of the files using the URL parameter.  
  - HTTPS: Type this value if the required installation files are stored at a secured URL. You must also provide the location of the files using the URL parameter, as well as the user name and password to |
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>access URL</td>
<td>access the URL using the UserName and Password parameters.</td>
</tr>
<tr>
<td>SourceLocation</td>
<td>Location of the folder that stores any required installation files. Type the location of the local file path. If you store the files in a local folder, do not provide a location in the URL parameter.</td>
</tr>
<tr>
<td>URL</td>
<td>Location of the URL for the HTTP or HTTPS location that stores any required installation files. Type the URL for the location that stores any required installation files. If you store the files at an HTTP or HTTPS location, do not provide a location in the SourceLocation parameter.</td>
</tr>
<tr>
<td>UserName</td>
<td>If you retrieve the installation files from a URL location, type a user name that has access to the URL location. If there is no login required to the URL or you retrieve the installation files from a local folder, you can leave this field blank.</td>
</tr>
<tr>
<td>Password</td>
<td>If you retrieve the installation files from a URL location, type a password for the user name. If there is no login required to the URL or you retrieve the installation files from a local folder, you can leave this field blank.</td>
</tr>
</tbody>
</table>

**Topology Configuration**

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ServicesRegConfig]</td>
<td>Section that specifies the services registration configuration.</td>
</tr>
<tr>
<td>hideDialog</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>multipleMachineEnvironment</td>
<td>TRUE or FALSE. FALSE sets this machine as a server node in the environment. Otherwise, specify the server</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ServersInCluster =</td>
<td>Provide the services registration server hostnames, separated by semicolons. The input must to be identical for all configured machines in this environment. It is recommended to configure an odd number of servers in a cluster. Example: server1.domain.com; server2.domain.com; server3.domain.com.</td>
</tr>
<tr>
<td>ConfigLoadBalancer =</td>
<td>TRUE or FALSE. FALSE ignores the rest of the values in this section of the file and the load balancer is not configured.</td>
</tr>
<tr>
<td>WebJSPServerLB =</td>
<td>Enter the location of the Web JSP server to use for load balancing. Example: http(s)://&lt;hostname&gt;:&lt;port&gt;/MicroStrategy</td>
</tr>
<tr>
<td>WebASPServerLB =</td>
<td>Enter the location of the Web ASP server to use for load balancing. Example: http(s)://&lt;hostname&gt;:&lt;port&gt;/WebVirtualDirector/asp/</td>
</tr>
<tr>
<td>MobileJSPServerLB =</td>
<td>Enter the location of the Mobile JSP server to use for load balancing. Example: http(s)://&lt;hostname&gt;:&lt;port&gt;/MicroStrategyMobile</td>
</tr>
<tr>
<td>MobileASPServerLB =</td>
<td>Enter the location of the Mobile ASP server to use for load balancing. Example: http(s)://&lt;hostname&gt;:&lt;port&gt;/MobileVirtualDirectory/asp/</td>
</tr>
<tr>
<td>LibraryWebMobileLB =</td>
<td>Enter the location of the Web Mobile server to use for load balancing. Example: http(s)://&lt;hostname&gt;:&lt;port&gt;/MicroStrategyLibrary</td>
</tr>
<tr>
<td>CollaborationServerLB =</td>
<td>Enter the location of the Collaboration server to use for load balancing. Example: http(s)://&lt;hostname&gt;:&lt;port&gt;/MicroStrategyIdentity</td>
</tr>
<tr>
<td>UsherServerLB =</td>
<td>Enter the location of the MicroStrategy Identity Server to use for load balancing. Example: http(s)://&lt;hostname&gt;:&lt;port&gt;/MicroStrategyIdentity</td>
</tr>
</tbody>
</table>
### Options and Configuration Guide

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s)://&lt;hostname&gt;[:&lt;port&gt;]</td>
<td></td>
</tr>
<tr>
<td>UsherNetworkManagerLB =</td>
<td>Enter the location of the MicroStrategy Identity Manager server to use for load balancing. Example: http(s)://&lt;hostname&gt;[:&lt;port&gt;]/networkmanager</td>
</tr>
</tbody>
</table>

#### Platform Analytics Configuration

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[PlatformAnalyticsConfig]</td>
<td>Section that specifies the configuration for Platform Analytics.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>PlatformAnalyticsUseCustomDB =</td>
<td>TRUE or FALSE. For non-upgrades, TRUE specifies a custom MySQL database to store Platform Analytics data. For upgrades, TRUE specifies a new MySQL database location.</td>
</tr>
<tr>
<td>PlatformAnalyticsDBHost =</td>
<td>The hostname for the MySQL database.</td>
</tr>
<tr>
<td>PlatformAnalyticsDBPort =</td>
<td>The port for the MySQL database.</td>
</tr>
<tr>
<td>PlatformAnalyticsDBUser =</td>
<td>The username for accessing the MySQL database.</td>
</tr>
<tr>
<td>PlatformAnalyticsDBPassword =</td>
<td>The user’s password for accessing the MySQL database.</td>
</tr>
<tr>
<td>PlatformAnalyticsOverwriteDB =</td>
<td>TRUE or FALSE. Set this option to overwrite an existing database with the same name. The default is FALSE.</td>
</tr>
</tbody>
</table>
## MicroStrategy Telemetry Cluster Configuration

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessagingServicesConfig</td>
<td>Section that configures MicroStrategy Telemetry.</td>
</tr>
<tr>
<td>HideDialog = TRUE or FALSE</td>
<td>FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>MessagingServicesCluster TRUE or FALSE</td>
<td>TRUE enables MicroStrategy Telemetry. The default is FALSE.</td>
</tr>
<tr>
<td>MessagingServicesLocalNode</td>
<td>Enter the local node’s hostname.</td>
</tr>
<tr>
<td>MessagingServicesRemoteNodes</td>
<td>Enter a semicolon separated list of the remote node’s hostnames.</td>
</tr>
</tbody>
</table>

## MicroStrategy Identity Configuration

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[UsherConfig]</td>
<td>Section that configures MicroStrategy Identity Services.</td>
</tr>
<tr>
<td>HideDialog = TRUE or FALSE</td>
<td>FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>ExpressSkipUsherConfig= TRUE or FALSE</td>
<td>TRUE to skip MicroStrategy Identity configuration in Express Installation. Set to FALSE when ExpressMode= FALSE.</td>
</tr>
<tr>
<td>CACertificateChain = [path]</td>
<td>Complete certificate chain for your SSL Server Certificate that you obtained from you IT Administrator. The path must be specified in an absolute format such as C:\folder\example.pem.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ServerCertificate =</td>
<td>The SSL Server Certificate (.crt) file for your Windows server. The path must be specified in an absolute format such as C:\folder\example.crt.</td>
</tr>
<tr>
<td>ServerCertificateKey =</td>
<td>The key file for your SSL Server Certificate (.crt). The path must be specified in an absolute format such as C:\folder\example.key.</td>
</tr>
<tr>
<td>ServerCertificateKeyPasswordFile =</td>
<td>If your CA-signed certificate has a password, create a text file containing this password and enter the text file location.</td>
</tr>
<tr>
<td>SMTPServer =</td>
<td>SMTP Server used for MicroStrategy Identity Badge email service.</td>
</tr>
<tr>
<td>SMTPServerPort =</td>
<td>SMTP Server Port.</td>
</tr>
<tr>
<td>SMTPUser =</td>
<td>If your server is password protected, enter the username for the server. This is optional.</td>
</tr>
<tr>
<td>SMTPUserPassword =</td>
<td>If your server is password protected, enter the password for the server. This is optional.</td>
</tr>
<tr>
<td>SMTPEmail =</td>
<td>The email address that is authorized to send emails from your SMTP server. This email address is used to send badge invitations for your MicroStrategy Identity Badge Network.</td>
</tr>
<tr>
<td>FQDN =</td>
<td>The Fully Qualified Domain Name of your Windows server.</td>
</tr>
</tbody>
</table>
## Open Source Software Agreement Dialog

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[OpenSourceSoftwareDialog]</td>
<td>MySQL is required as a repository in Express Install or required by MicroStrategy Identity products in Custom Install. R is used by Intelligence Server to process R based functions to enable R analytics</td>
</tr>
<tr>
<td>Agreed</td>
<td><strong>TRUE</strong> or <strong>FALSE</strong>. <strong>FALSE</strong> displays the dialog box. The default is <strong>FALSE</strong>. <strong>TRUE</strong> to allow the installer to download the open source software components for you. <strong>FALSE</strong> to follow the links below to download the open source components yourself, being sure to save all of the components to your Downloads folder. If not all the components are placed correctly in your Downloads folder, the Express Install or Custom Install cannot proceed. Please note that if you choose <strong>TRUE</strong>, you are authorizing the installer to download MySQL and R analytics components on your behalf. These MySQL and R analytics components are open-source software provided under the GPL licenses. These components are not provided by MicroStrategy. For access to the source code for these components, please visit visit <a href="http://www.mysql.com">http://www.mysql.com</a>, <a href="https://www.r-project.org/">https://www.r-project.org/</a> and related links. <strong>MySQL</strong>: <a href="http://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.28-winx64.zip">http://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.28-winx64.zip</a> <strong>MySQL Connector with ODBC 5.3.4</strong>:</td>
</tr>
</tbody>
</table>
### MySQL Connector Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ServerDefinitionSetting]</td>
<td>Section specifying whether MicroStrategy Intelligence Server will use the server definition included with the Tutorial.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>OverwriteServerDefinition =</td>
<td>TRUE or FALSE. This option relates to the Tutorial. Set this option to overwrite existing MicroStrategy Intelligence Server definitions from a previous install. The default is FALSE.</td>
</tr>
</tbody>
</table>
### Analytics Module

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[AnalyticsSetting]</td>
<td>Section that specifies the DSN used to connect to the MicroStrategy Analytics Modules.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>OverwriteDSN =</td>
<td>TRUE or FALSE. Set this option to overwrite an existing DSN with the same name. The data source names are as follows:</td>
</tr>
<tr>
<td></td>
<td>• Analytics_Metadata</td>
</tr>
<tr>
<td></td>
<td>• CAM_WH_AC</td>
</tr>
<tr>
<td></td>
<td>• FRAM_WH_AC</td>
</tr>
<tr>
<td></td>
<td>• HRAM_WH_AC</td>
</tr>
<tr>
<td></td>
<td>• MicroStrategy_Tutorial_Data</td>
</tr>
<tr>
<td></td>
<td>• SAM_WH_AC</td>
</tr>
<tr>
<td></td>
<td>• SDAM_WH_AC</td>
</tr>
<tr>
<td></td>
<td>The default is FALSE.</td>
</tr>
</tbody>
</table>

### MicroStrategy Web Virtual Directory

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[WebVirtualDirectory]</td>
<td>Section that specifies the virtual directory to be used for the MicroStrategy Web application.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>VirtualDirectory =</td>
<td>Enter a name for the virtual directory. The default is MicroStrategy.</td>
</tr>
<tr>
<td>RemoveVD =</td>
<td>YES or NO. This option is for the uninstall only. Set</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>[MobileVirtualDirectory]</td>
<td>Section that specifies the virtual directory to be used for the MicroStrategy Mobile Server applications.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>VirtualDirectory =</td>
<td>Enter a name for the virtual directory. The default is MicroStrategyMobile.</td>
</tr>
<tr>
<td>ReconfigureVirtualDirectory =</td>
<td>TRUE or False. This option is relevant to upgrading MicroStrategy from a pre-9.0.1m version. TRUE replaces the virtual directory used to support MicroStrategy Mobile for BlackBerry with the new virtual directory specified for MicroStrategy Mobile Server. For more information on upgrade installations, see the Upgrade Guide.</td>
</tr>
<tr>
<td>RemoveVD =</td>
<td>YES or NO. This option is for the uninstall only. Set this option to remove an existing MicroStrategy Mobile Server virtual directory from a previous installation. The default is NO.</td>
</tr>
</tbody>
</table>
### MicroStrategy MDX Cube Provider Virtual Directory

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[MDXCubeProviderVirtualDirectory]</td>
<td>Section that specifies the virtual directory to be used for the MicroStrategy MDX Cube Provider.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>VirtualDirectory =</td>
<td>Enter a name for the virtual directory. The default is MicroStrategyMDX.</td>
</tr>
<tr>
<td>ReconfigureVirtualDirectory =</td>
<td>TRUE or FALSE. Set this option to TRUE if the virtual directory for the MicroStrategy MDX Cube Provider should be reconfigured to support a new virtual directory.</td>
</tr>
<tr>
<td>RemoveVD =</td>
<td>YES or NO. This option is for the uninstall only. Set this option to remove an existing MicroStrategy MDX Cube Provider virtual directory from a previous installation. The default is NO.</td>
</tr>
</tbody>
</table>

### Subscription Portal Virtual Directory

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[PortalVirtualDirectory]</td>
<td>Section that specifies the virtual directory to be used for MicroStrategy Subscription Portal.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>VirtualDirectory =</td>
<td>Enter a name for the virtual directory. The default is NarrowcastServer.</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RemoveVD =</strong> YES or NO. This option is for the uninstall only. Set this option to remove an existing MicroStrategy Subscription Portal virtual directory from a previous installation. The default is NO.</td>
</tr>
</tbody>
</table>

### MicroStrategy Web Services Virtual Directory

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[WebServicesDirectory]</strong></td>
<td>Section that specifies the virtual directory to be used for MicroStrategy Web Services.</td>
</tr>
<tr>
<td><strong>HideDialog =</strong></td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td><strong>VirtualDirectory =</strong></td>
<td>Enter a name for the virtual directory. The default is MicroStrategyWS.</td>
</tr>
<tr>
<td><strong>RemoveVD =</strong></td>
<td>YES or NO. This option is for the uninstall only. Set this option to remove an existing MicroStrategy Subscription Portal virtual directory from a previous installation. The default is NO.</td>
</tr>
</tbody>
</table>

### MicroStrategy Office Web Services URL

This information applies to MicroStrategy Office, the add-in for Microsoft Office applications which is no longer actively developed. It was substituted with a new add-in, MicroStrategy for Office, which supports Office 365 applications. The initial version does not yet have all the functionalities of the previous add-in.

For more information, see the MicroStrategy for Office page in the 2019 Update 1 Readme and the MicroStrategy for Office Online Help.
In allation and Configura tion Guide

## Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HideDialog</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>AllowBlankURL</td>
<td>TRUE or FALSE. Specify whether to allow a blank URL. The installation routine validates the provided URL. If no URL is provided, the user is informed that it has been left blank and needs to be configured with the MicroStrategy Office Configuration Tool. If this is set to TRUE, the user message is not displayed if the URL is left blank. The default is FALSE.</td>
</tr>
<tr>
<td>URL</td>
<td>Enter a URL pointing to a valid MicroStrategy Web Services installation, for example, <a href="http://localhost/MicroStrategyWS/MSTRWS.asmx">http://localhost/MicroStrategyWS/MSTRWS.asmx</a></td>
</tr>
</tbody>
</table>

### MicroStrategy Office Setting

This information applies to MicroStrategy Office, the add-in for Microsoft Office applications which is no longer actively developed.

It was substituted with a new add-in, MicroStrategy for Office, which supports Office 365 applications. The initial version does not yet have all the functionalities of the previous add-in.

For more information, see the MicroStrategy for Office page in the 2019 Update 1 Readme and the MicroStrategy for Office Online Help.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[MSOfficeLoadOptions]</td>
<td>Section specifying the options that determine if the MicroStrategy Office toolbar is loaded in the installed</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HideDialog = TRUE</td>
<td>Microsoft Office applications. True or False. False displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>ConfigureExcel = TRUE</td>
<td>TRUE or FALSE. Specify to load the MicroStrategy Office toolbar by default when the Microsoft Excel application runs. This applies only if Excel is installed in the target machine. The default is TRUE.</td>
</tr>
<tr>
<td>ConfigureWord = TRUE</td>
<td>TRUE or FALSE. Specify to load the MicroStrategy Office toolbar by default when the Microsoft Word application runs. This applies only if Word is installed on the target machine. The default is TRUE.</td>
</tr>
<tr>
<td>ConfigurePowerpoint = TRUE</td>
<td>TRUE or FALSE. Specify to load the MicroStrategy Office toolbar by default when the Microsoft PowerPoint application runs. This applies only if PowerPoint is installed on the target machine. The default is TRUE.</td>
</tr>
</tbody>
</table>

### Intelligence Server Service Account

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[IServerServiceAccount]</td>
<td>Section specifying the Windows account for the MicroStrategy Intelligence Server service. You have two options:</td>
</tr>
<tr>
<td></td>
<td>• bypass entering the account information</td>
</tr>
<tr>
<td></td>
<td>• enter the account information</td>
</tr>
<tr>
<td>HideDialog = TRUE</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>SkipAccountSetting = TRUE</td>
<td>TRUE or FALSE. Set TRUE to bypass the service account setting in the MicroStrategy Intelligence Server Setting dialog box. If you bypass it, then</td>
</tr>
</tbody>
</table>
### Options | Description
--- | ---
ServiceStartup | AUTO or MANUAL. Select to set the Intelligence Server service startup to be automatic or manual. The default is AUTO.

#### Narrowcast Server Service Account

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[NarrowcastServiceAccount]</td>
<td>Section specifying the Windows account from which the MicroStrategy Narrowcast Server service will run.</td>
</tr>
<tr>
<td>HideDialog</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>SkipAccountSetting</td>
<td>TRUE or FALSE. If you specify this value as FALSE, the service account settings are not skipped and the MicroStrategy Narrowcast Server setting dialog box is displayed. Specify the details of the Windows account that the MicroStrategy Narrowcast Server services will use to log on and click Next to proceed with the installation process.</td>
</tr>
<tr>
<td>Domain</td>
<td>Enter the domain where the account is located.</td>
</tr>
<tr>
<td>Login</td>
<td>Enter the user name of the account to use.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password for the account.</td>
</tr>
</tbody>
</table>
Start Copying Files Dialog

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Summary]</td>
<td>Section that specifies the installation summary in the Start Copying Files dialog box.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
</tbody>
</table>

MicroStrategy Installation Wizard Complete Dialog

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Finish]</td>
<td>Section that specifies the MicroStrategy Installation Wizard Complete dialog box.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
</tbody>
</table>

Example of a response.ini file for Custom Installation to install all components

Starting from 10.4, you can find the sample_custom.ini in the same location as the installation setup.exe. Replace any text between the angled brackets <> with your own specific information.

```ini
# MicroStrategy 11 Response File
#
# This file contains entries targeted to Custom Mode, which will install and configure the full platform on one machine.
#
# For more information on Silent Install please refer to the section on "Installing and configuring with a response.ini file" in the Installation and Configuration guide available at
```

© 2019, MicroStrategy Inc.
We are working to improve our Silent and Automated installation use-cases. Feedback on Silent or Automated installations is welcomed via the MicroStrategy Community (Platform Services > Platform > Secure Enterprise - Windows) or via Technical Support to the Deployment Team.

Here is a direct link (as of Jun 2017) to the Community sub-section on platform deployment:


# Usage

In an Administrator Command Line window (Windows Button > CMD, Right Click and Run As Administrator)

Silent Install Usage: #PathToSetupExe# --responseFile="#PathToResponseIni#" -s -f1#PathToSetupIss# -f2#PathToLogFile#

Example: C:\Setup.exe --responseFile="C:\response_custom.ini" -s -f1C:\Setup.iss -f2C:\Setup.log

This assumes the following:

Setup.exe is located at: C:\Setup.exe
MicroStrategy Response File is located at: C:\response_custom.ini
Installer Setup.iss is located at: C:\Setup.iss
Output Log file should be written to: C\Setup.log

Response entries start here, replace any text between angled brackets (<>) with your own text

[Installer]
ExpressMode=FALSE
PropertiesFilesOverwrite=FALSE
EnableTracing=FALSE
HideAllDialogs=TRUE

# After initial installation is finished, choose whether to automatically reboot the machine.
# A reboot is required.
# TRUE  - Indicates the machine will automatically reboot (recommended)
# FALSE - Indicates that no automatic reboot will be performed. Task Manager (setup.exe) or the Install.log may be used to determine once the installation is finished as no other indication will be provided.
ForceReboot=TRUE
PreventReboot=FALSE
CheckTCPIP=TRUE
CheckIIS=TRUE
CreateShortcuts=TRUE
CheckRenameOperations=TRUE
AnalyticsOverwrite=True
TutDeliveryOverwrite=True
BackupFiles=FALSE
RunConfigWizard=FALSE
StopAllServices=TRUE
StopIIS=TRUE
EnableASPServices=TRUE
EnableASPNETServices=TRUE
ShowWelcomeScreen=FALSE
EnterpriseManagerOverwrite=TRUE
###ConfigWizardResponseFile=Response.ini

# Path and File Name for the Installation Log file.
# If no value is specified, the default location will be used:
C:\Program Files (x86)\Common Files\MicroStrategy\install.log
LogFile=

[Welcome]
HideDialog=TRUE
RemoveAll=FALSE

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
# Customer Information

# Please specify your first name, last name, email address, the name of the company
# for which you work and the license key.

# If MicroStrategy Identity Badge is installed, this information is used to create
# the initial MicroStrategy Identity Badge network and user badge.

[UserRegistration]
HideDialog=TRUE

# First name of user
UserFirstName=<value>

# Last name of user
UserLastName=<value>

# Email address of user
UserEmail=<value>

# Company
CompanyName=<value>

# License key
LicenseKey=<value>

[LicenseDetail]
HideDialog=TRUE

[SetupExpress]
HideDialog=TRUE

# Product Install Location.
# Use the following values to specify the install location for MicroStrategy products.
# The default path of TargetDirectory is C:\Program Files (x86)\MicroStrategy
# The default path of COMMONFILES is C:\Program Files (x86)\Common Files\MicroStrategy

[SuiteTarget]
HideDialog=TRUE
TargetDirectory=

[InitialPaths]
COMMONFILES =

[ComponentSelection]
HideDialog=TRUE

## MicroStrategy Product Visibility State

## For GUI based installations, this section provides a visibility state setting for every MicroStrategy product.
## Legal values are:
## TRUE - Indicates that the product is visible for selection or deselection
## FALSE - Indicates that the product is not visible for selection or deselection

## The default value for visibility state settings is "true".
## For example, to make "MicroStrategy Intelligence Server" visible
## for selection or deselection
## IServerVisible=TRUE

## All licensed products are visible by default, so in case you want to
prevent
# them from being visible, make sure they are set to false.

DeveloperVisible=TRUE
AnalystVisible=TRUE
ArchitectVisible=TRUE
ServerAdminVisible=TRUE
FunctionPluginVisible=FALSE
CommandManagerVisible=TRUE
EnterpriseManagerVisible=TRUE
ObjectManagerVisible=TRUE
IntegrityManagerVisible=TRUE
IServerVisible=TRUE
IServerOLAPServicesVisible=TRUE
IServerReportServicesVisible=TRUE
IServerDistributionServicesVisible=TRUE
IServerTransactionServicesVisible=TRUE
WebAnalystVisible=TRUE
WebProfessionalVisible=TRUE
WebReporterVisible=TRUE
WebServerASPNETVisible=TRUE
WebServerJSPVisible=TRUE
WebServicesASPNETVisible=TRUE
WebServicesJSPVisible=TRUE
OfficeVisible=TRUE
MobileVisible=TRUE
MobileClientVisible=TRUE
MobileServerASPSVisible=TRUE
MobileServerJSPVisible=TRUE
AnalyticsModulesVisible=TRUE
NCSAdminVisible=TRUE
DeliveryEngineVisible=TRUE
SubscriptionPortalVisible=TRUE
TutorialDeliveryInstallVisible=TRUE
TutorialDeliveryConfigureVisible=TRUE
SequeLinkVisible=TRUE
PortletsVisible=TRUE
MDXCubeProviderVisible=TRUE
GISConnectorsVisible=TRUE
SystemManagerVisible=TRUE
# New in 10.3
UsherServerVisible=TRUE
UsherNetworkManagerVisible=TRUE
UsherProfessionalVisible=TRUE
# New in 10.5
MessagingServicesVisible=TRUE
TomcatVisible=TRUE
MySQLVisible=TRUE
# New in 10.8
RVisible=TRUE
# New in 10.9
LibraryWebMobileVisible=TRUE
CollaborationServerVisible=TRUE
# New in 11.0
DataServerVisible=TRUE
CommunityConnectorsVisible=TRUE
PlatformAnalyticsVisible=TRUE
CertificateManagerVisible=TRUE

# MicroStrategy Product Selection State
# Determines which products are installed (GUI and Silent). A selection state setting is available for every MicroStrategy product.
# Legal values are:
#
#   TRUE - Indicates that the product is selected for installation
#   FALSE - Indicates that the product is not selected for installation
#
# The default value for Selection state settings is "true".
#
# For example, to select "MicroStrategy Intelligence Server"
# for installation use
#
# IServerSelect=TRUE
Most licensed products are selected by default, so in case you want to prevent them from being installed make sure they are unselected too.

DeveloperSelect=TRUE
AnalystSelect=TRUE
ArchitectSelect=TRUE
ServerAdminSelect=TRUE
FunctionPluginSelect=FALSE
CommandManagerSelect=TRUE
EnterpriseManagerSelect=TRUE
ObjectManagerSelect=TRUE
IntegrityManagerSelect=TRUE
IServerSelect=TRUE
IServerOLAPServicesSelect=TRUE
IServerReportServicesSelect=TRUE
IServerDistributionServicesSelect=TRUE
IServerTransactionServicesSelect=TRUE
WebAnalystSelect=TRUE
WebProfessionalSelect=TRUE
WebReporterSelect=TRUE
WebServerASPNETSelect=TRUE
WebServerJSPSelect=TRUE
WebServicesASPNETSelect=TRUE
WebServicesJSPSelect=TRUE
OfficeSelect=TRUE
MobileSelect=TRUE
MobileClientSelect=TRUE
MobileServerASPSelect=TRUE
MobileServerJSPSelect=TRUE
AnalyticsModulesSelect=TRUE
NCSAdminSelect=TRUE
DeliveryEngineSelect=TRUE
SubscriptionPortalSelect=TRUE
TutorialDeliveryInstallSelect=TRUE
TutorialDeliveryConfigureSelect=TRUE
SequeLinkSelect=TRUE
PortletsSelect=TRUE
MDXCubeProviderSelect=TRUE
GISConnectorsSelect=TRUE
SystemManagerSelect=TRUE
# New in 10.3
UsherServerSelect=TRUE
UsherNetworkManagerSelect=TRUE
UsherProfessionalSelect=TRUE
# New in 10.5
MessagingServicesSelect=TRUE
TomcatSelect=TRUE
MySQLSelect=TRUE
# New in 10.8
RSelect=TRUE
# New in 10.9
LibraryWebMobileSelect=TRUE
CollaborationServerSelect=TRUE
# New in 11.0
DataServerSelect=TRUE
CommunityConnectorsSelect=TRUE
PlatformAnalyticsSelect=TRUE
CertificateManagerSelect=TRUE

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
# Topology Configuration
# The Workstation Topology helps administrators to monitor and manage services
# in a MicroStrategy environment. Choose your configuration below.

[ServicesRegConfig]
HideDialog=TRUE

# If MultipleMachineEnvironment=FALSE, this machine will be set as a server node in the environment.
# Otherwise, please specify server nodes in ServersInCluster option.
MultipleMachineEnvironment=FALSE

# The input below will be entered identically for all configured machines in this environment.
# Example: servername1.acme.com; servername2.acme.com; servername3.acme.com

# It is recommended to configure an odd number of servers in a cluster
ServersInCluster=

# If ConfigLoadBalancer=FALSE, the rest of the values in this section will be ignored and the load balancer will NOT be configured
ConfigLoadBalancer=FALSE

WebJSPServerLB=http(s)://<hostname>:[port]/MicroStrategy
WebASPServerLB=http(s)://<hostname>:[port]/<WebVirtualDirectory>/asp/
MobileJSPServerLB=http(s)://<hostname>:[port]/MicroStrategyMobile
MobileASPServerLB=http(s)://<hostname>:[port]/<MobileVirtualDirectory>/asp/
LibraryWebMobileLB=http(s)://<hostname>:[port]/MicroStrategyLibrary
CollaborationServerLB=http(s)://<hostname>:[port]
UsherServerLB=http(s)://<hostname>:[port]
UsherNetworkManagerLB=http(s)://<hostname>:[port]/networkmanager

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## 
## ## 
# Platform Analytics Configuration
[PlatformAnalyticsConfig]
HideDialog=TRUE

# For non-upgrade, specify custom MySQL database to store Platform Analytics data
# For upgrade, specify a new MySQL database location
PlatformAnalyticsUseCustomDB=FALSE
PlatformAnalyticsDBHost=
PlatformAnalyticsDBPort=
PlatformAnalyticsDBUser=
PlatformAnalyticsDBPassword=

# If set as TRUE, existing schema will be overwritten. Otherwise, existing schema will be kept and upgraded if needed
PlatformAnalyticsOverwriteDB=

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## 
## ## 
# MicroStrategy Telemetry Cluster Configuration
[MessagingServicesConfig]
HideDialog=TRUE
MessagingServicesCluster=FALSE

# Please provide local node's hostname
MessagingServicesLocalNode=

# Please provide a semicolon separated list of remote nodes' hostname
MessagingServicesRemoteNodes=

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## #### ## ##

# MicroStrategy Identity Product Configuration
# The parameters in [UsherConfig] section only apply when MicroStrategy
# Identity products are selected.
[UsherConfig]
HideDialog=TRUE

# For Custom install, ExpressSkipUsherConfig has to be FALSE for
# installation to proceed
# For Express install, if ExpressSkipUsherConfig=TRUE, the rest of the
# values in this section will be ignored and MicroStrategy Identity will
# NOT be configured
ExpressSkipUsherConfig=FALSE

# Path and File Name for the Certificate Bundle (Chain), typically
# .pem. e.g. C:\certs\labs.pem
CACertificateChain=<value>

# Path and File Name for the Server Certificate. e.g. C:\certs\labs.crt
ServerCertificate=<value>

# Path and File Name for the Server Certificate Key file. e.g.
# C:\certs\labs.key
ServerCertificateKey=<value>

# Optional: Path and File Name to the file which contains the password
# to the Server Certificate Key (.key) file.
ServerCertificateKeyPasswordFile=
# SMTP Server name
SMTPServer=<value>

# SMTP Server port number
SMTPServerPort=<value>

# Optional: SMTP Server Using SSL
SMTPUseSSL=FALSE

# Optional: SMTP User name
SMTPUser=

# Optional: SMTP user password
SMTPUserPassword=

# SMTP email
SMTPEmail=<value>

# Fully Qualified Domain Name
FQDN=<value>

# New in 10.8
[OpenSourceSoftwareDialog]
HideDialog=TRUE

# Agreement to download Open Source Software Installation Files
# MicroStrategy Identity Services require MySQL
# R is used by Intelligence Server to process 'R' based functions to enable 'R' Analytics
# TRUE - Indicates you agree for the installation to automatically download the required open source software installation files on your behalf.
# FALSE - Indicates the open source software files are pre-downloaded in the User's Downloads folder. (Typically C:\Users<user>\Downloads)
# Default is FALSE
AgreeToDownloadOpenSourceSoftware=
[ServerDefinitionSetting]
HideDialog=TRUE
OverwriteServerDefinition=FALSE

[AnalyticsSetting]
HideDialog=TRUE
OverwriteDSN=FALSE

[WebVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategy
ReconfigureVirtualDirectory=TRUE

[MobileVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyMobile
ReconfigureVirtualDirectory=TRUE

[PortalVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=NarrowcastServer
ReconfigureVirtualDirectory=TRUE

[WebServicesVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyWS
ReconfigureVirtualDirectory=TRUE

[OfficeWebServicesURL]
HideDialog=TRUE
AllowBlankURL=FALSE
URL=http://localhost/MicroStrategyWS/mstrws.asmx

[MSOfficeLoadOptions]
HideDialog=TRUE
ConfigureExcel=TRUE
ConfigurePowerpoint=TRUE
Your license key determines which MicroStrategy components will be available for your installation. For example, if your license key does not include MicroStrategy OLAP Services, then you cannot use

\texttt{IServerOLAPServicesSelect=TRUE} and \texttt{IServerOLAPServicesVisible=TRUE} to install these components.

Example of a \texttt{response.ini} file for Express Installation

Starting from 10.4 you can find the \texttt{sample_express.ini} in the same location as the installation \texttt{setup.exe}. Replace any text between angled brackets \texttt{<>} with your own specific information.
## MicroStrategy 11 Response File ##

This file contains a subset of entries required for Express Mode, which will install and configure the full platform on one machine. For more information on Express Mode please refer to the section on "Performing a MicroStrategy Express Installation" in the Installation and Configuration guide available at https://www2.microstrategy.com/producthelp/manuals/en/InstallationConfig.pdf.

Additional entries are not included in this file and may be optionally included for greater flexibility. For a full list, please refer to the "Installing and configuring with a response.ini file" section in the Installation and Configuration guide.

We are working to improve our Silent and Automated installation use-cases. Feedback on Silent or Automated installations is welcomed via the MicroStrategy Community (Platform Services > Platform > Secure Enterprise - Windows) or via Technical Support to the Deployment Team. Here is a direct link (as of Jun 2017) to the Community sub-section on platform deployment:


### Usage ###

In an Administrator Command Line window (Windows Button > CMD, Right Click and Run As Administrator)

Silent Install Usage: #PathToSetupExe --responseFile="#PathToResponseIni#" -s -f1#PathToSetupIss# -f2#PathToLogFile#

Example: C:\Setup.exe --responseFile="C:\response_express.ini" -s -f1C:\Setup.iss -f2C:\Setup.log

The above example assumes the following:
# Setup.exe is located at: C:\Setup.exe
# MicroStrategy Response File is located at: C:\response_express.ini
# Installer Setup.iss is located at: C:\Setup.iss
# Output Log file should be written to: C:\Setup.log
#
# Response entries start here, replace any text between angled brackets
<() with your own text

[Installer]
ExpressMode=TRUE
HideAllDialogs=TRUE

# After initial installation is finished, choose whether to automatically reboot the machine.
# A reboot is required.
# TRUE - Indicates the machine will automatically reboot (recommended)
# FALSE - Indicates that no automatic reboot will be performed. Task Manager (setup.exe) or the Install.log may be used to determine once the installation is finished as no other indication will be provided.
ForceReboot=TRUE
CheckIIS=TRUE
RunConfigWizard=FALSE
StopAllServices=TRUE
StopIIS=TRUE
ShowWelcomeScreen=FALSE

# Path and File Name for the Installation Log file.
# If no value is specified, the default location will be used:
LogFile=

# Product Install Location.
# Use the following values to specify the install location for MicroStrategy products.
# The default path of TargetDirectory is C:\Program Files
(x86)\MicroStrategy
# The default path of COMMONFILES is C:\Program Files (x86)\Common
# Files\MicroStrategy

[SuiteTarget]
HideDialog = TRUE
TargetDirectory =

[InitialPaths]
COMMONFILES =

[Welcome]
HideDialog=TRUE
RemoveAll=FALSE

# Customer Information
#
# Please specify your first name, last name, email address, the name of
# the company for which you work and the license key.
#
# For Express Installation, this information is used to create the
# initial MicroStrategy Identity Badge network and user badge.

[UserRegistration]
HideDialog=TRUE

# First name of user
UserFirstName=<value>

# Last name of user
UserLastName=<value>

# Email address of user
UserEmail=<value>

# Company
CompanyName=<value>

# License key
LicenseKey=<value>

[LicenseDetail]
HideDialog=TRUE

[SetupExpress]
HideDialog=TRUE

##
##
##
##
##
##
##
##
##
##
##
##
##
##
# MicroStrategy Identity Badge configuration
#
# MicroStrategy Identity Badge requires SSL Certificates and an SMTP server to send badge invitations.

[UsherConfig]
HideDialog=TRUE

# For Express install, select whether MicroStrategy Identity Services will be configured.
# TRUE - All other values in this [UsherConfig] section will be ignored, MicroStrategy Identity Services will be installed, but MicroStrategy Identity will NOT be configured.
# FALSE - Values in this section are required and will be used to install and configure MicroStrategy Identity.
ExpressSkipUsherConfig=<value>

# Path and File Name for the Certificate Bundle (Chain), typically .pem. e.g. C:\certs\labs.pem
CACertificateChain=<value>

# Path and File Name for the Server Certificate, typically .crt. e.g. C:\certs\labs.crt
ServerCertificate=<value>
# Path and File Name for the Server Certificate Key file, typically .key. e.g. C:\certs\labs.key
ServerCertificateKey=<value>

# Optional: Path and File Name to the file which contains the password to the Server Certificate Key (.key) file.
ServerCertificateKeyPasswordFile=

# SMTP Server name
SMTPServer=<value>

# SMTP Server port number
SMTPServerPort=<value>

# Optional: SMTP Server Using SSL
SMTPUseSSL=FALSE

# Optional: SMTP User name
SMTPUser=

# Optional: SMTP user password
SMTPUserPassword=

# SMTP Sender email address
SMTPEmail=<value>

# Fully Qualified Domain Name
FQDN=<value>

# New in 10.8
[OpenSourceSoftwareDialog]
HideDialog=TRUE

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
# Agreement to download Open Source Software Installation Files
# MicroStrategy Identity Services require MySQL
# R is used by Intelligence Server to process 'R' based functions to
Using the response.ini file

The setup program supports several command-line parameters. The following applies to this function:

- Parameters using double dashes, such as `--auto`, are defined by MicroStrategy. For example, you can use the `--auto` parameter as follows:

  ```
  Path\setup.exe --Auto=TRUE --
  LogFile="C:\install.log"
  ```

- The command line is not case-sensitive.

The following parameters are supported by the setup program:

- `Auto=` instructs the setup program to use the response file and default values to enable a one-click installation. If a component, such as serial key or disk space has an invalid value, the setup program automatically reverts to multiple-click mode, and all dialog boxes are displayed.
- **ResponseFile**= contains responses to installation questions and redefines default parameters. The path and file name must be in double quotes (" "). If you use this parameter, do not use any other parameters.

- **ConfigFile**= used by the Configuration Wizard to set up a repository, a server, or a client. The path and file name must be in double quotes (" ").

- **LogFile**= used to specify an alternative location and/or name (other than install.log) for the log file in the Common Files directory. If only the file name is entered, the default location remains the Common Files directory. Once specified, the alternative file becomes the default.

---

**To use the response.ini file to install MicroStrategy components**

1. **Save your response.ini file to the directory C:\.** You can save to a different directory, but the example command provided in these steps assumes the response file is saved to the directory location C:\.

2. **From the Windows Start menu, select Programs, then select Accessories, and then right-click Command Prompt and select Run as Administrator.** The User Account Control dialog box opens.

   ![Information Icon]

   The steps to open a Windows command prompt with administrator privileges may be different depending on your version of Windows.

3. **Click Yes to open the command prompt with administrator privileges.** The command prompt is displayed.

4. **Type the following command in the Windows command line:**

   ```
   Path\setup.exe --ResponseFile="C:\response.ini"
   ```
Where *Path* is the directory where the setup.exe file is stored.

5. Click **Enter**.

### Configuring your Installation with a response.ini File

The Configuration Wizard walks you through the process of setting up the environment for the MicroStrategy products installed in your system. It is possible to configure server definition, project source names, and the metadata repository using a `response.ini` file. The steps required to create a `response.ini` file to configure MicroStrategy are provided in *Configuring MicroStrategy with a Response File*, page 404.

### Uninstalling with a response.ini File

You can uninstall all MicroStrategy products at once using a `response.ini` file. You must create a response file with the `RemoveAll` parameter set to `TRUE` in the Welcome section. This is also known as a silent uninstallation.

⚠️ You must save the file as ANSI encoding.

Before uninstallation begins, the MicroStrategy application:

- Checks for user privileges. If they are not valid, uninstallation stops.
- Checks for running components. If one is found, uninstallation stops.
- Stops and deletes the MicroStrategy Intelligence Server service.
- Deletes application created files.
Example of a response.ini file to uninstall MicroStrategy

You can use the following response file to remove all MicroStrategy products:

```
[Installer]
PropertiesFilesOverwrite=FALSE
EnableTracing=FALSE
HideAllDialogs=TRUE
ForceReboot=TRUE
PreventReboot=FALSE
CheckTCPIP=TRUE
CheckIIS=TRUE
CheckSP=TRUE
CreateShortcuts=TRUE
CheckRenameOperations=TRUE
AnalyticsOverwrite=FALSE
TutDeliveryOverwrite=FALSE
BackupFiles=FALSE
RunConfigWizard=FALSE
StopAllServices=TRUE
StopIIS=TRUE
EnableASPServices=FALSE
ConfigWizardResponseFile=
RegistrySizeReq=
LogFile=

[Welcome]
HideDialog=TRUE
RemoveAll=TRUE

[WebVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategy
RemoveVD=YES

[MobileVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyMobile
RemoveVD=YES

[OperationsManagerVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyOM
RemoveVD=YES

[PortalVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=NarrowcastServer
RemoveVD=YES
```
[WebServicesDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyWS
RemoveVD=YES
[Finish]
HideDialog=TRUE

For details on creating a response.ini file, see Creating a response.ini file, page 560.

After you have created a response.ini file, open a Windows command prompt to uninstall all MicroStrategy products. From the Windows Start menu, select Programs, then select Accessories, and then right-click Command Prompt and select Run as Administrator. The User Account Control dialog box opens.

The steps to open a Windows command prompt with administrator privileges may be different depending on your version of Windows.

Click Yes to open the command prompt with administrator privileges. The command prompt is displayed. Type the following script at the command prompt to uninstall all MicroStrategy products:

Path1\setup.exe --ResponseFile= "Path2\response.ini"

Where the Path1 for setup.exe must be the path to the original setup.exe used to install MicroStrategy products. The Path2 for the response file is the path where you saved your response.ini file.

Silent Installation

A silent, or unattended, installation is one that presents no user interface. Silent installations are useful for system administrators who do not want users to interfere with the installation. They are typically implemented by IT departments that perform package-based installations across the network.
You can use silent installation to easily embed MicroStrategy products with other applications. This can be done to develop an OEM application that includes MicroStrategy functionality. For information on deploying a silent installation for OEM applications, see *OEM Silent Installations, page 688*.

**Silent Installation Output**

The system verifies compliance with installation prerequisites and places related messages in a file created for that purpose. The following applies to the generation and storage of output messages during silent installation:

- **The** `MSTRInst.log` **file** is created and placed in the Temp folder.
- **The** `MSTRInst.log` **file** is maintained during the entire setup.
- All system-generated messages, including messages containing reasons for pre-installation termination, are stored in the `MSTRInst.log` **file**.

  If there are installation termination messages in subsequent installation instances that use the same dialog flow, they are also stored in the `MSTRInst.log`.

**The table below identifies the information that the** `MSTRInst.log` **file** includes:

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function ([F])</td>
<td>Identifies the function calls in the setup script.</td>
</tr>
<tr>
<td>Information ([I])</td>
<td>Logs information about the setup that is running.</td>
</tr>
<tr>
<td>Warning ([W])</td>
<td>Includes feedback that you must verify related to the setup. For</td>
</tr>
</tbody>
</table>
example, in normal mode, when MicroStrategy applications are running on a machine where the setup is being run, you are prompted to close all MicroStrategy applications before proceeding. In silent mode, you are not prompted, and instead the setup terminates.

Severe [S] Includes fatal problems that prevent the setup from proceeding. For example, the Intelligence Server Service cannot be created and setup fails as a result.

A typical line in the MSTRInst.log file includes source file name, function name, source line number, and time. It appears as follows:

```
[Z:\InstallMaster\Setup\Script Files\MALicense.rul]
[UseDLL][l: 28][1318179ms][W] Le file C:\WINDOWS\TEMP\{84D0D5E2-719A-11D7-847C-000C293B5019}\{B339F3B3-E78C-45E8-B4D2-3C46C1C13663}\MAInst.dll couldn't be loaded in memory.
```

When reviewing warning messages in the MSTRInst.log file, look for [W] and [S] to find related problems.

Activating a Silent Installation

After the silent installation is complete, you must activate the MicroStrategy installation within 30 days. To activate the installation you can follow the instructions provided in Chapter 5, Activating Your Installation.

Troubleshooting Silent Installations

Silent installation may not work if you are installing in a different environment than the one you recorded. This is the case because any dialog box that was not recorded previously is not recognized, such
as a prompt to stop your Intelligence Server or your IIS Web server. If this happens, verify:

- The version of Intelligence Server to ensure that you have the right one for the products you are installing
- You do not have any MicroStrategy applications running
- The `setup.log` file to see if the `ResultCode=0`
- The `install.log` for any recorded errors during installation and the `MSTRInst.log` file for any possible errors. The most common errors are:
  - -8, which is an invalid path to the MicroStrategy Installation Wizard silent response `.iss` file.
  - -12, which is dialog boxes out of order. This occurs because an unrecorded screen opened when running the silent install.

It is recommended that you create a silent install that can be used with a `response.ini` file. For more information, see OEM Silent Installations, page 688. This way you can change the settings in the `response.ini` file without having to generate a new `.iss` file.

Silent Installation of MicroStrategy Office

This information applies to MicroStrategy Office, the add-in for Microsoft Office applications which is no longer actively developed.

It was substituted with a new add-in, MicroStrategy for Office, which supports Office 365 applications. The initial version does not yet have all the functionalities of the previous add-in.

For more information, see the MicroStrategy for Office page in the 2019 Update 1 Readme and the MicroStrategy for Office Online Help.
You can also install MicroStrategy Office as its own stand-alone installation, which lets you install only MicroStrategy Office. The stand-alone installation of MicroStrategy Office can also be used to install updates of the MicroStrategy Office product.

To perform a silent installation of MicroStrategy Office as a stand-alone installation, refer to the procedure below.

You can also perform an installation of MicroStrategy Office as its own stand-alone installation, but with the assistance of a MicroStrategy Office installation wizard. The steps to use the MicroStrategy Office installation wizard are provided in the MicroStrategy for Office Online Help.

You must have Microsoft .NET Framework and Microsoft Web Services Enhancement Runtime to install and support MicroStrategy Office. For the required versions of these components, see the MicroStrategy Readme.

- You must have Microsoft Windows Installer 4.5 to install MicroStrategy Office.
- Microsoft Office must already be installed on the machine.

To perform a silent installation of MicroStrategy Office

1. Download the MicroStrategy Office stand-alone installation files. You can retrieve these from the MicroStrategy installation disk, or from the MicroStrategy download site. The MicroStrategy Office stand-alone installation files include files named MicroStrategyOffice.msi and MicroStrategyOffice64.msi. These .msi files are for installing MicroStrategy Office on 32-bit and 64-bit versions of Microsoft Office, respectively.
2. From the Windows Start menu, go to Programs > Accessories, and then right-click Command Prompt and select Run as Administrator. The User Account Control dialog box opens.

   - The steps to open a Windows command prompt with administrator privileges may be different depending on your version of Windows.

3. Click Yes to open the command prompt with administrator privileges. The command prompt is displayed.

4. From a command prompt, navigate to the MicroStrategy Office update installation folder.

5. To view information on the options to run the silent install command, enter the following command:

   `msiexec.msi /?`

   This includes information on options to log the installation details to a log file.

6. To run the silent installation using installation options defined by the current MicroStrategy Office configuration, enter one of the following commands:

   - To install on 32-bit versions of Microsoft Office:

     `msiexec.exe /i MicroStrategyOffice.msi /qn`

   - To install on 64-bit versions of Microsoft Office:

     `msiexec.exe /i MicroStrategyOffice64.msi /qn`

   - You can also use additional parameters as part of the silent installation command, including the following:

     - **INSTALLDIR**: Defines the directory in which MicroStrategy Office will be installed. For example, you can define this
parameter as INSTALLDIR="C:\Program Files\MicroStrategy\Office".

- **WSURL**: Defines the URL for MicroStrategy Web Services. For example, you can define this parameter as WSURL="http://localhost/MicroStrategyWS/MSTRWS.asmx".

- **LW**: Specify whether to load the MicroStrategy Office toolbar by default when the Microsoft Word application runs. This applies only if Word is installed in the target machine. The default is 3, which loads the toolbar. You can set this option to 2 to not load the toolbar for Microsoft Word.

- **LE**: Specify whether to load the MicroStrategy Office toolbar by default when the Microsoft Excel application runs. This applies only if Excel is installed in the target machine. The default is 3, which loads the toolbar. You can set this option to 2 to not load the toolbar for Microsoft Excel.

- **LP**: Specify whether to load the MicroStrategy Office toolbar by default when the Microsoft PowerPoint application runs. This applies only if PowerPoint is installed in the target machine. The default is 3, which loads the toolbar. You can set this option to 2 to not load the toolbar for Microsoft PowerPoint.

- **OFFICECREATESHORTCUTS**: Specify whether to create a shortcut for MicroStrategy Office in the Windows Start menu. The default is 1, which creates the shortcut. You can set this option to 0 to exclude the creation of this shortcut.

**Configuring your MicroStrategy Installation**

After completing the steps to install Intelligence Server, you can continue the set up and configuration of your installation. To help guide the rest of your installation and configuration steps, refer to the
section *Installation and Configuration Checklists, page 110* in *Chapter 1, Planning Your Installation*, for installation and configuration checklists.
AUTOMATED INSTALLATION ON LINUX
This section explains the various methods of performing a fully automated and unattended installation within the MicroStrategy platform when you do not have access to a Linux graphical user interface (GUI).

Intelligence Server configurations possible through the command line on Linux are covered in Chapter 12, Configuring MicroStrategy Using Command Line Tools.

Before installing MicroStrategy products, see Chapter 1, Planning Your Installation for important preinstallation information.

Silent Installation

A silent or unattended installation is one that presents no graphical user interface (GUI). Silent installations allow you to automate the installation, so it can be called from a script and executed without user interaction. Silent installations are useful for system administrators who do not want users to run the installation themselves. The silent installation can be done on one or more computers.

Completing a Silent Installation

To run silent installation, you must create an options file and then run it with the MicroStrategy Installation Wizard. Save the options file as `options.txt`. An example of an options.txt file is provided in Example of an options file, page 626 below. You can use the example as a template and replace italicized text with your own information.

This options file or response file is used with the command line argument `-options` to modify the wizard settings. The settings that can be specified for the wizard are listed in the next section, Parameters for a Silent Installation, page 651.
1. Log on to the computer where you are installing one or more MicroStrategy products.

2. You can access the installation files by asking your system administrator to share the files in a network location.


4. Open options.txt in a text editor.

5. Specify a value for a setting by replacing the character's Value. For detailed information on the parameters and values that can be supplied with the options.txt file for a silent installation, see Parameters for a Silent Installation, page 651.

6. Save the changes to the options.txt file.

7. To use the options file on a silent installation, specify -silent -options FileName as a command line argument to the wizard, where FileName is the name of this options file, for example, options.txt. For example, type the command:

        setup.sh -silent -options options.txt

Example of an options file

Copy and paste this example to create an options.txt file. Make sure you check for correct spaces and new lines in all file paths.

For descriptions of each of the options listed in this example, see Parameters for a Silent Installation, page 651.

The example below assumes you are using the full set of installation files to support the MicroStrategy installation. If you have downloaded only a subset of the .tzp files that are required for the MicroStrategy installation, you must define where these files are stored, using the
Parameters described in Install on Demand Options, page 677.

```
### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ######  ```
Here is a direct link (as of June 2017) to the Community sub-section on platform deployment:


Silent Installation.

The options file is required for silent installations, specify -silent
-options <file-name> as a command line arguments to the wizard,
where
<file-name> is the name of this options file.

For example:

setup.sh -silent -options options.txt

System Requirements Warnings

The silent installation will install MicroStrategy 10 even if some system
requirements are not completely meet.
Please read the release notes file and configure your system so it
meets the
requirements before running this setup and check the installation log after
# Installation

The installation is complete.

# Installation Properties

Please specify general installation properties.

# MicroStrategy License Agreement Default Selection

In the MicroStrategy Installation Wizard, it is legally required to accept the MicroStrategy License Agreement in order to proceed with the desired installation operation. Use this option to specify the default selected option selected for the License Agreement Dialog. Legal values are:

- true - Accept the License Agreement
- false - Do not accept the License Agreement

If no option is provided, 'reject' is used as default.

### licenseAgreement.accept=<value>

# Customer Information

Please specify your name, the name of the company for which you work
and the
# license key.

# User
### userRegistration.user=<value>

# Company
### userRegistration.company=<value>

# License Key
### userRegistration.cdKey=<value>

# MicroStrategy Installation Selection
# For a new Installation, use "new" (i.e. install.Instance/new)
# For an existing Installation, specify its home path.
# Home path is a directory that identifies an installation and stores
# configuration files and application launchers (suite.homeDirectory in
# MicroStrategy 10 Install Locations section). You may refer to <home>
tag under
# <suite>MstrSuite</suite> in mstrinstall.xml.
#
# For example
#
# install.Instance=/home/user/MicroStrategy10
#
### install.Instance=<value>

# MicroStrategy Operation to perform on instance
# For a new Installation, Legal values are:
#   FRESH_INSTALL - Indicates a new installation will be performed
#
# For an existing Installation, Legal values are:
#   MODIFY - Indicates the installation will be modified.
#   REPAIR - Indicates the installation will be repaired.
#   UNINSTALL - Indicates the installation will be uninstalled.
#   UPGRADE - Indicates the installation will be upgraded
#   HOTFIX_UNINSTALL - Indicates a hotfix package will be removed
#
# For example
#
#   install.Operation=MODIFY
#
### install.Operation=<value>

### Install Locations

# The install locations of the product. Specify a valid directories into which
# the product should be installed.
# It is recommended to specify a different directory for each one of the install
# locations:
#
# Home - This directory will store configuration files and application launchers
# Install - This directory will store all the binaries and other static files
# Log - Application logs will be created here.
#
# NOTE: Special characters (!,%,$,^,\,?,#) and space characters are not valid. All
# special characters will be ignored.
### suite.homeDirectory=<value>

### suite.installDirectory=<value>

### suite.logDirectory=<value>

##
# MicroStrategy Product Selection State
#
# A selection state setting is available for every MicroStrategy product.
# Legal values are:
#
#   true - Indicates that the product is selected for installation
#   false - Indicates that the product is not selected for installation
#
# The default value for Selection state settings is "true".
#
# For example, to select "MicroStrategy Intelligence Server Universal Edition"
# for installation use
#
#   IntelligenceServer.active=true
#
# All licensed products are selected by default, so in case you want to
# prevent them from being installed make sure they are unselected too.

##
# MicroStrategy Intelligence Server Universal Edition
# MicroStrategy Intelligence Server Universal Edition will be automatically selected for installation if any of the following products are selected for installation:
"MicroStrategy Report Services Universal",
"MicroStrategy OLAP Services Universal"
"MicroStrategy Distribution Services"
"MicroStrategy Transaction Services"

# New in 10.8
# R
### R.active=<value>

# MicroStrategy Web Universal
# MicroStrategy Web Universal Reporter will be automatically selected for installation if MicroStrategy Web Universal Analyst is selected.

© 2019, MicroStrategy Inc.
# MicroStrategy Web Universal Reporter & Analyst will be automatically selected
# for installation if MicroStrategy Web Universal Professional is selected.

# MicroStrategy Web Universal Reporter
### WebReporter.active=<value>

# MicroStrategy Web Universal Analyst
### WebAnalyst.active=<value>

# MicroStrategy Web Universal Professional
### WebProfessional.active=<value>

# MicroStrategy Portlets
### Portlets.active=<value>

# MicroStrategy GIS Connectors
### GISConnectors.active=<value>

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## 634
## MicroStrategy Community Connectors

### CommunityConnectors.active=<value>

## MicroStrategy Command Manager

### CommandManager.active=<value>

## MicroStrategy System Manager

### SystemManager.active=<value>

## MicroStrategy Platform Analytics

### PlatformAnalytics.active=<value>

## MicroStrategy Enterprise Manager

### EnterpriseManager.active=<value>
# Migration

## MicroStrategy Integrity Manager

```bash
### IntegrityManager.active=<value>
```

## MicroStrategy Usher Security Services

```bash
### UsherSecurityServer.active=<value>
```

## Usher Network Manager

```bash
### UsherNetworkManager.active=<value>
```

## Usher Mobile

```bash
### UsherMobile.active=<value>
```

## MicroStrategy Messaging Services

```bash
### MessagingServices.active=<value>
```

## New in 10.9

## MicroStrategy Library Web & Mobile

```bash
### LibraryWebMobile.active=<value>
```
In allation and Configuration Guide

## New in 10.9

# MicroStrategy Collaboration Server

### CollaborationServer.active=<value>

## New in 11.0

# MicroStrategy Certificate Manager

### CertificateManager.active=<value>

## Product Install Location.

# Use the following values to specify the install location for individual products.

# Warning: Special (!,%,$,^,\,?,#) and space characters are not valid.

# characters will be ignored.

# Web Universal

### WebUniversal.deployDirectory=<value>

# MicroStrategy Portlets

### Portlets.installDirectory=<value>

# MicroStrategy GIS Connectors

### GISConnectors.installDirectory=<value>
# MicroStrategy Community Data Connectors
### CommunityConnectors.installDirectory=<value>

# MicroStrategy Web Services J2EE
### WebServices.installDirectory=<value>

# MicroStrategy Mobile Server JSP
### MobileServer.installDirectory=<value>

# MicroStrategy Command Manager
### CommandManager.installDirectory=<value>

# MicroStrategy System Manager Install Location
### SystemManager.installDirectory=<value>

# MicroStrategy Platform Analytics Install Location
### PlatformAnalytics.installDirectory=<value>

# Usher Security Server
### UsherSecurityServer.installDirectory=<value>

# Usher Network Manager
### UsherNetworkManager.installDirectory=<value>

# Usher Mobile
### UsherMobile.installDirectory=<value>

## # CPU License Information
## # This value should be specified when the license being used for
## MicroStrategy
## # Intelligence Server Universal Edition is based on CPUs. Legal values
## are integers
## # between 1 and either the number of CPUs allowed by the license or the
number of CPUs in the machine, whichever is lower.

For example to set the number of CPUs to 2, use:

```
cpuCount.number=2
```

To set this options to the maximum value allowed, use:

```
cpuCount.number=maximum
```

The default value for this setting is "maximum"

```
## cpuCount.number=<value>
```

Agreement to download Open Source Software Installation Files

R is used by Intelligence Server to process 'R' based functions to enable 'R' Analytics

true - Indicates you agree for the installation to automatically download the required open source software installation files on your behalf.
false - Installation cannot proceed. You must deselect the Open Source Software.

Default is false

```
## AgreeToDownloadOpenSourceSoftware=<value>
```

Skip R reinstall on repair

By default R will be reinstalled on repair. Set this option to true
if you
# don't want to reinstall R on repair.
# true  - Don't reinstall R on repair
# false - Reinstall R on repair
### SkipReinstallR=<value>

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
## ## ## ## 
# MicroStrategy Messaging Services Cluster Configuration
# 
# Important: If this machine is part of a clustered environment (i.e. 2
# or more Intelligence Server
# nodes) MicroStrategy recommends clustering Messaging Services too.
# 
# This increases the reliability, scalability and performance of
Platform Analytics. To cluster
# Messaging Services set the property
'MessagingServices.cluster.enable' to 'true' and provide the
# hostname of at least 3 Messaging Services nodes. One in the
'MessagingServices.cluster.localnode'
# property and the rest in a comma separated list in
'MessagingServices.cluster.remotenodes'.
#
# Please note that Messaging Services is installed by default when you
install and Intelligence
# Server or Platform Analytics.

# Create a Messaging Services cluster for Platform Analytics
#  true  - Creates a Messaging Services cluster
#  false - Configures Messaging Services as a single node.
# The default value is false.
### MessagingServices.cluster.enable=<value>

# Local node's hostname, do not use the value 'localhost' or any
loopback IP address (127.0.0.1).
# The default value is output of the Java call InetAddress.getLocalHost
().getHostName().
### MessagingServices.cluster.localnode=<value>

# Comma separated list of the remote nodes' hostnames. Provide at least two.
# For example: node.mydomain.com, othernode.mydomain.com
### MessagingServices.cluster.remotenodes=<value>

### Usher Security Server Information

# Please specify the information required for the Usher Security Server configuration.

# Tomcat directory location for Usher Security Server
### UsherSecurityServer.tomcatDir=<value>

# Enable SSL for Usher Security Server database connection.
# Valid values are "true" or "false". The default value is "false".
### UsherSecurityServer.serverDBenableSsl=<value>

# Usher Security Server installs a database which is a system of record for
# individual Usher identities. Please provide the configuration parameters for
# the Usher Security Server to communicate with the database with MySQL
# Community Edition 5.0 or higher.
### UsherSecurityServer.serverDBHost=<value>
### UsherSecurityServer.serverDBPort=<value>
### UsherSecurityServer.serverDBUser=<value>
### UsherSecurityServer.serverDBPassword=<value>
### UsherSecurityServer.serverDBInstance=<value>
### UsherSecurityServer.serverLogDBInstance=<value>

# For new installations use this option to indicate if any existing schema should be
# dropped or preserved.
# true - Drop the schema and create a fresh configuration.
# false - Preserve the schema and upgrade it if necessary.
# The default value is false.
### UsherSecurityServer.overwriteDb=<value>

# Please specify the information required for setting up a trust-relationship for
# Usher Security Server using the Public-key Infrastructure (PKI).
# Server (one-way SSL) authentication only - Port
### UsherSecurityServer.serverPortOne=<value>

# Client and Server (two-way SSL) mutual authentication - Port
### UsherSecurityServer.serverPortTwo=<value>

# Public key SSL certificate file (.crt)
### UsherSecurityServer.serverSslCert=<value>

# Private key file (.key)
### UsherSecurityServer.serverSslKey=<value>

# Certification Authority's (CA) certificate file (.pem)
### UsherSecurityServer.serverCaCert=<value>

# Please specify the information required for setting up a trust-relationship for
# Agent Gateway using the Public-key Infrastructure (PKI).
# Agent Gateway (one-way SSL) authentication only - Port
### UsherSecurityServer.gatewayPort=<value>

# Use this option to specify if you want to use the same SSL certificate as Usher
# Security Server.
# true - Use the same SSL certificates
# false - Do not use the same SSL certificates
### UsherSecurityServer.gatewayUseSameCert=<value>

# Public key SSL certificate file (.crt)
### UsherSecurityServer.gatewaySslCert=<value>
# Private key file (.key)
### UsherSecurityServer.gatewaySslKey=<value>

# Certification Authority's (CA) certificate file (.pem)
### UsherSecurityServer.gatewayCaCert=<value>

# Usher Network Manager Information

# Please specify the information required for the Usher Network Manager configuration.

# Apache directory location and user credentials for Usher Network Manager
### UsherNetworkManager.apacheDir=<value>
### UsherNetworkManager.apacheUser=<value>
### UsherNetworkManager.apacheGroup=<value>

# Usher Network Manager installs a database to manage Usher identities. Please provide the configuration parameters for the Usher Network Manager to communicate with the database.

# Use this option to specify if you want to use the same connection information as Usher Security Server.
# true - Use the same connection
# false - Do not use the same connection
### UsherNetworkManager.useSameDBSetting=<value>

### UsherNetworkManager.DBHost=<value>
### UsherNetworkManager.DBPort=<value>
### UsherNetworkManager.DBUser=<value>
### UsherNetworkManager.DBPassword=<value>
# Please provide a Database instance name for Usher Network Manager.
### UsherNetworkManager.DBInstance=<value>

# For new installations use this option to indicate if any existing database instance
# should be dropped or preserved.
# true   - Drop the schema and create a fresh configuration.
# false  - Preserve the schema and upgrade it if necessary.
# The default value is false.
### UsherNetworkManager.overwriteDb=<value>

# Enable SSL for Usher Network Manager database connection.
# Valid values are "true" or "false". The default value is "false".
### UsherNetworkManager.DBenableSsl=<value>

# Use this option to specify if you want to use the same SSL certificate as Usher
# Security Server.
# true   - Use the same SSL certificates
# false  - Do not use the same SSL certificates
### UsherNetworkManager.dbUseSameCert=<value>

# Public key SSL certificate file (.crt)
### UsherNetworkManager.dbSslCert=<value>

# Private key file (.key)
### UsherNetworkManager.dbSslKey=<value>

# Certification Authority's (CA) certificate file (.pem)
### UsherNetworkManager.dbCaCert=<value>

## MicroStrategy Library Web & Mobile Information
# Please specify the information required for the MicroStrategy Library Web &
# Mobile configuration.

# Automatically deploy MicroStrategy Library Web and Mobile.
# Valid values are "true" or "false". The default value is "true".
### LibraryWebMobile.deploy=<value>
# Tomcat directory location for MicroStrategy Library Web & Mobile
### LibraryWebMobile.tomcatDir=<value>

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
# MicroStrategy Platform Analytics Information
# Platform Analytics installs a database to store telemetry from the
# MicroStrategy
# platform. Please provide the database connection parameters.
### PlatformAnalytics.DBHost=<value>
### PlatformAnalytics.DBPort=<value>
### PlatformAnalytics.DBUser=<value>
### PlatformAnalytics.DBPassword=<value>
### PlatformAnalytics.overwriteDb=<value>
### PlatformAnalytics.upgrade.useDifferentDb=<value>

## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
# Topology Configuration
# The Workstation Topology helps administrators to monitor and manage
# services
# in a MicroStrategy environment. Choose your configuration below.

# Indicate cluster environment.
# SINGLE - indicates single machine environment
# MULTIPLE - indicates multiple machine environment
The default value for this setting is "SINGLE"

```bash
### ServicesRegistration.machine.environment=<value>
```

The input below will be entered identically for all configured machines in this environment.

```bash
### ServicesRegistration.serverHosts=<value>
```

### Load balancer for MicroStrategy Mobile Server

Example: http://localhost/MicroStrategyMobile

```bash
### WebServer.loadBalancer=<value>
### MobileServer.loadBalancer=<value>
### LibraryWebMobile.loadBalancer=<value>
### CollaborationServer.loadBalancer=<value>
### UsherServer.loadBalancer=<value>
### UsherNetworkManager.loadBalancer=<value>
```

```bash
## Registering as a service

Register MicroStrategy processes as a service so that processes automatically start up after system startup.

Legal values are:

```bash
true
false
```

Note: Only root users may register processes as a service.

```bash
### RegisterServices=<value>
```

Hide Select Components dialog box

Use this option to completely hide the product selection dialog during the
# execution of the MicroStrategy Installation Wizard. This option can
# be used
# in combination with individual product visibility/state options to
# have better
# control on the products to be installed/removed. If the dialog is not
# visible,
# and no specific visibility/state options have been specified, default
# values
# will be used.
# Legal values for this option are:
#
#    true - Indicates that the Select Components dialog box will be
#           visible
#    false - Indicates that the Select Components dialog box will be
#            hidden
#
# The default value for this setting is "true".

# SelectComponents.visible=<value>

### MicroStrategy Product Visibility

# In the MicroStrategy Installation Wizard, the Select Components
dialog box
# contains check boxes to select or clear for products to be installed.
You
# can specify whether you want the following products to be visible to
the user.
# Legal values are:
#
#    true - Indicates that the product is selected for installation
#    false - Indicates that the product is not selected for
#             installation
#
# The default value for Selection state settings is "true".
# If a product is not visible, it will not be installed, # independently of the value specified for its selection state (see the # Product Selection State section).
#
# MicroStrategy Intelligence Server Universal Edition
### IntelligenceServer.visible=<value>

# MicroStrategy Report Services Universal
### ReportServices.visible=<value>

# MicroStrategy OLAP Services Universal
### OLAPServices.visible=<value>

# MicroStrategy Distribution Services
### DistributionServices.visible=<value>

# MicroStrategy Transaction Services
### TransactionServices.visible=<value>

# New in 10.8
# R
### R.visible=<value>

# MicroStrategy Web Universal Reporter
### WebReporter.visible=<value>

# MicroStrategy Web Universal Analyst
### WebAnalyst.visible=<value>

# MicroStrategy Web Universal Professional
### WebProfessional.visible=<value>

# MicroStrategy Portlets
### Portlets.visible=<value>

# MicroStrategy GIS Connectors
### GISConnectors.visible=<value>
# MicroStrategy Web Services J2EE
### WebServices.visible=<value>

# MicroStrategy Mobile Server JSP
### MobileServer.visible=<value>

# MicroStrategy Community Data Connectors
### CommunityConnectors.visible=<value>

# MicroStrategy Command Manager
### CommandManager.visible=<value>

# MicroStrategy Integrity Manager
### IntegrityManager.visible=<value>

# MicroStrategy System Manager
### SystemManager.visible=<value>

# MicroStrategy Platform Analytics
### PlatformAnalytics.visible=<value>

# MicroStrategy Enterprise Manager
### EnterpriseManager.visible=<value>

# Usher Security Server
### UsherSecurityServer.visible=<value>

# Usher Network Manager
### UsherNetworkManager.visible=<value>

# Usher Mobile
### UsherMobile.visible=<value>

# MicroStrategy Messaging Services
### MessagingServices.visible=<value>

# New in 10.9
# MicroStrategy Library Web & Mobile
### LibraryWebMobile.visible=<value>
# New in 10.9
# MicroStrategy Collaboration Server
### CollaborationServer.visible=<value>

# New in 11.0
# MicroStrategy Certificate Manager
### CertificateManager.visible=<value>

### Install On Demand Options
### The kind of remote location. The valid values are "FileSystem", "HTTP" or "HTTPS"
### InstallOnDemand.style=<value>
### The path to the remote location. Set this value if InstallOnDemand.style is set to "FileSystem"
### InstallOnDemand.sourceLocation=<value>
### The URL for HTTP or HTTPS styles. Set this value if InstallOnDemand.style is set to HTTP or HTTPS
### InstallOnDemand.url=<value>
### Use this setting to skip the certificate checking when using HTTPS. Valid values are "true" or "false"
### InstallOnDemand.bypassCertificateChecking=<value>
### The user name for HTTP or HTTPS styles. Set this value if InstallOnDemand.style is set to HTTP or HTTPS
# and the connection requires user name and password
### InstallOnDemand.user=<value>
### InstallOnDemand.password=<value>

Parameters for a Silent Installation

The following parameters define how a silent installation is performed.

The settings follow their descriptions, in the format:

```
settingname=Value
```

License Agreement

Define whether the license agreement is accepted by default:

```
licenseAgreement.accept=Value
```

You can define this parameter with one of the following values:

- **true**: The MicroStrategy license agreement is accepted by default. If you perform the installation as a silent installation that does not display the MicroStrategy Installation Wizard, you must use this value to install MicroStrategy successfully.

- **false**: The MicroStrategy license agreement is not accepted by default. The user that installs MicroStrategy must select to accept the license agreement to continue with the MicroStrategy installation.

Customer Information

Your name, the name of the company for which you work, and the license key:
- User:
  ```
  userRegistration.user=Value
  ```

- Company:
  ```
  userRegistration.company=Value
  ```

- License key:
  ```
  userRegistration.cdKey=Value
  ```

**MicroStrategy Installation Instance**

You can either install a new instance of MicroStrategy, or modify an existing MicroStrategy installation:

- To install a new instance of MicroStrategy:
  ```
  install.Instance=new
  ```

- To modify an existing MicroStrategy installation:
  ```
  install.Instance=InstallPath
  ```

**MicroStrategy Operations**

In addition to installing MicroStrategy, you can also modify, repair, uninstall, and upgrade existing MicroStrategy installations:

- To install a new instance of MicroStrategy:
  ```
  install.Operation=FRESH_INSTALL
  ```

- To modify an existing MicroStrategy installation:
  ```
  install.Operation=MODIFY
  ```
To repair an existing MicroStrategy installation by performing the previous installation attempt:

```bash
install.Operation=REPAIR
```

To uninstall an existing MicroStrategy installation:

```bash
install.Operation=UNINSTALL
```

To upgrade an existing MicroStrategy installation:

```bash
install.Operation=UPGRADE
```

To install a MicroStrategy Update installation:

```bash
install.Operation=HOTFIX_INSTALL
```

To uninstall an existing MicroStrategy Update installation:

```bash
install.Operation=HOTFIX_UNINSTALL
```

MicroStrategy Install Locations

The install locations of the product. Specify valid directories where the product should be installed. For additional requirements when choosing directories, see *Installing with the MicroStrategy Installation Wizard for Linux, page 165*.

- Home directory: The location where the MicroStrategy configuration files and application launchers are to be installed.

  ```bash
  suite.homeDirectory=Path
  ```

- Install directory: The location where the MicroStrategy products are to be installed.

  ```bash
  suite.installDirectory=Path
  ```
- Log directory: The location where the MicroStrategy application logs are to be created.

  ```
silent.logDirectory=Path
  ```

**Product Features**

When you install products using an options file, the following two values may be specified for each product:

- A visible option, which can use one of the following values:
  - **true**: Indicates that the feature is displayed in the MicroStrategy Installation Wizard as available for installation.
  - **false**: Indicates that the feature is not displayed in the MicroStrategy Installation Wizard as available for installation. If you define a product's visible option as false, it cannot be installed.

  If no value is specified, the default is **true** for all products. You can also define the visible option for all products using the parameter `SelectComponents.visible`. You can exclude these visible options for each product if you are using the options file as part of a completely silent installation where no user interface is displayed to the user.

- An active option, which can use one of the following values:
  - **true**: Indicates that the feature is selected for installation.
  - **false**: Indicates that the feature is not selected for installation, or the product is to be uninstalled as part of an installation that modifies or uninstalls previously installed MicroStrategy software.

  To review a description of each MicroStrategy project, see *MicroStrategy Products and Components, page 11.*
MicroStrategy Intelligence Server

- The state of whether MicroStrategy Intelligence Server is displayed in the MicroStrategy Installation Wizard:
  \[
  \text{IntelligenceServer.visible}=\text{Value}
  \]

- The selection state of MicroStrategy Intelligence Server.
  \[
  \text{IntelligenceServer.active}=\text{Value}
  \]

To select MicroStrategy Intelligence Server for installation, use:

\[
\begin{align*}
\text{IntelligenceServer.visible} &= \text{true} \\
\text{IntelligenceServer.active} &= \text{true}
\end{align*}
\]

MicroStrategy Intelligence Server is installed if you choose to install any of the following MicroStrategy products:

- \textit{MicroStrategy Report Services, page 655}
- \textit{MicroStrategy OLAP Services, page 656}
- \textit{MicroStrategy Distribution Services, page 656}
- \textit{MicroStrategy Transaction Services, page 657}

MicroStrategy Report Services

- The state of whether MicroStrategy Report Services is displayed in the MicroStrategy Installation Wizard:
  \[
  \text{ReportServices.visible}=\text{Value}
  \]

- The selection state of MicroStrategy Report Services.
  \[
  \text{ReportServices.active}=\text{Value}
  \]
For example, to select MicroStrategy Report Services for installation, use:

```
ReportServices.visible=true
ReportServices.active=true
```

MicroStrategy OLAP Services

- The state of whether MicroStrategy OLAP Services is displayed in the MicroStrategy Installation Wizard:

```
OLAPServices.visible=Value
```

- The selection state of MicroStrategy OLAP Services.

```
OLAPServices.active=Value
```

- For example, to select MicroStrategy OLAP Services for installation, use:

```
OLAPServices.visible=true
OLAPServices.active=true
```

MicroStrategy Distribution Services

- The state of whether MicroStrategy Distribution Services is displayed in the MicroStrategy Installation Wizard:

```
DistributionServices.visible=Value
```

- The selection state of MicroStrategy Distribution Services.

```
DistributionServices.active=Value
```

- For example, to select MicroStrategy Distribution Services for installation, use:
DistributionServices.visible=true
DistributionServices.active=true

**MicroStrategy Transaction Services**

- The state of whether MicroStrategy Transaction Services is displayed in the MicroStrategy Installation Wizard:

  TransactionServices.visible=\textit{Value}

- The selection state of MicroStrategy Transaction Services.

  TransactionServices.active=\textit{Value}

- For example, to select MicroStrategy Transaction Services for installation, use:

  TransactionServices.visible=true
  TransactionServices.active=true

**RAnalytics**

- The state of whether MicroStrategy Transaction Services is displayed in the MicroStrategy Installation Wizard:

  R.visible=\textit{Value}

- The selection state of MicroStrategy Transaction Services.

  R.active=\textit{Value}

- For example, to select MicroStrategy Transaction Services for installation, use:

  R.visible=true
MicroStrategy Web Reporter

- The state of whether MicroStrategy Web Reporter is displayed in the MicroStrategy Installation Wizard:

```java
WebReporter.visible=Value
```

- The selection state of MicroStrategy Web Reporter.

```java
WebReporter.active=Value
```

- For example, to select MicroStrategy Web Reporter for installation, use:

```java
WebReporter.visible=true
WebReporter.active=true
```

MicroStrategy Web Analyst

- The state of whether MicroStrategy Web Analyst is displayed in the MicroStrategy Installation Wizard:

```java
WebAnalyst.visible=Value
```

- The selection state of MicroStrategy Web Analyst.

```java
WebAnalyst.active=Value
```

- For example, to select MicroStrategy Web Analyst for installation, use:

```java
WebAnalyst.visible=true
WebAnalyst.active=true
```
MicroStrategy Web Professional

- The state of whether MicroStrategy Web Professional is displayed in the MicroStrategy Installation Wizard:

  `WebProfessional.visible=\text{Value}`

- The selection state of MicroStrategy Web Professional.

  `WebProfessional.active=\text{Value}`

- For example, to select MicroStrategy Web Professional for installation, use:

  `WebProfessional.visible=true`
  `WebProfessional.active=true`

MicroStrategy Portlets

- The state of whether MicroStrategy Portlets is displayed in the MicroStrategy Installation Wizard:

  `Portlets.visible=\text{Value}`

- The selection state of MicroStrategy Portlets.

  `Portlets.active=\text{Value}`

- For example, to select MicroStrategy Portlets for installation, use:

  `Portlets.visible=\text{true}`
  `Portlets.active=\text{true}`
MicroStrategy GIS Connectors

- The state of whether MicroStrategy GIS Connectors is displayed in the MicroStrategy Installation Wizard:

```
GISConnectors.visible=Value
```

- The selection state of MicroStrategy GIS Connectors.

```
GISConnectors.active=Value
```

- For example, to select MicroStrategy GIS Connectors for installation, use:

```
GISConnectors.visible=true
GISConnectors.active=true
```

MicroStrategy Web Services J2EE

- The state of whether MicroStrategy Web Services J2EE is displayed in the MicroStrategy Installation Wizard:

```
WebServices.visible=Value
```

- The selection state of MicroStrategy Web Services J2EE.

```
WebServices.active=Value
```

- For example, to select MicroStrategy Web Services J2EE for installation, use:

```
WebServices.visible=true
WebServices.active=true
```
MicroStrategy Mobile Server JSP

- The state of whether MicroStrategy Mobile Server JSP is displayed in the MicroStrategy Installation Wizard:

```
MobileServer.visible=Value
```

- The selection state of Mobile Server JSP.

```
MobileServer.active=Value
```

- For example, to select MicroStrategy Mobile Server JSP for installation, use:

```
MobileServer.visible=true
MobileServer.active=true
```

MicroStrategy Community Connectors

- The state of whether MicroStrategy Community Connectors is displayed in the MicroStrategy Installation Wizard:

```
CommunityConnectors.active=Value
```

- For example, to select MicroStrategy Community Connectors for installation, use:

```
CommunityConnectors.visible=true
CommunityConnectors.active=true
```

MicroStrategy Command Manager

- The state of whether MicroStrategy Command Manager is displayed in the MicroStrategy Installation Wizard:
The selection state of MicroStrategy Command Manager.

For example, to select MicroStrategy Command Manager for installation, use:

```
CommandManager.visible=true
CommandManager.active=true
```

MicroStrategy System Manager

The state of whether MicroStrategy System Manager is displayed in the MicroStrategy Installation Wizard:

```
SystemManager.visible=Value
```

The selection state of MicroStrategy System Manager.

For example, to select MicroStrategy System Manager for installation, use:

```
SystemManager.visible=true
SystemManager.active=true
```

MicroStrategy Platform Analytics

The state of whether MicroStrategy System Manager is displayed in the MicroStrategy Installation Wizard:
PlatformAnalytics.visible=\text{Value}

- The selection state of MicroStrategy System Manager.

PlatformAnalytics.active=\text{Value}

- For example, to select MicroStrategy System Manager for installation, use:

PlatformAnalytics.visible=true

PlatformAnalytics.active=true

MicroStrategy Enterprise Manager

- The state of whether MicroStrategy Enterprise Manager is displayed in the MicroStrategy Installation Wizard:

EnterpriseManager.visible=\text{Value}

- The selection state of MicroStrategy Enterprise Manager.

EnterpriseManager.active=\text{Value}

- For example, to select MicroStrategy Enterprise Manager for installation, use:

EnterpriseManager.visible=true

EnterpriseManager.active=true

MicroStrategy Integrity Manager

- The state of whether MicroStrategy Integrity Manager is displayed in the MicroStrategy Installation Wizard:
The selection state of MicroStrategy Integrity Manager.

For example, to select MicroStrategy Integrity Manager for installation, use:

MicroStrategy Identity Server

The state of whether MicroStrategy Identity Server is displayed in the MicroStrategy Installation Wizard:

The selection state of MicroStrategy Identity Server.

For example, to select MicroStrategy Identity Server for installation, use:

MicroStrategy Identity Manager

The state of whether MicroStrategy Identity Manager is displayed in the MicroStrategy Installation Wizard:
- The selection state of MicroStrategy Identity Manager.

```
UsherNetworkManager.visible=Value
```

- For example, to select MicroStrategy Identity Manager for installation, use:

```
UsherNetworkManager.visible=true
UsherNetworkManager.active=true
```

### MicroStrategy Communicator

- The state of whether MicroStrategy Communicator is displayed in the MicroStrategy Installation Wizard:

```
UsherMobile.visible=Value
```

- The selection state of MicroStrategy Communicator.

```
UsherMobile.active=Value
```

- For example, to select MicroStrategy Communicator for installation, use:

```
UsherMobile.visible=true
UsherMobile.active=true
```

### MicroStrategy Messaging Services

- The state of whether MicroStrategy Messaging Services is displayed in the MicroStrategy Installation Wizard:
MessagingServices.visible=\texttt{Value}

- The selection state of MicroStrategy Messaging Services.

MessagingServices.active=\texttt{Value}

- For example, to select MicroStrategy Messaging Services for installation, use:

MessagingServices.visible=\texttt{true}

MessagingServices.active=\texttt{true}

MicroStrategy Library (Web and Mobile)

- The state of whether MicroStrategy Library is displayed in the MicroStrategy Installation Wizard:

LibraryWebMobile.visible=\texttt{Value}

- The selection state of MicroStrategy Library.

LibraryWebMobile.active=\texttt{Value}

- For example, to select MicroStrategy Library for installation, use:

LibraryWebMobile.visible=\texttt{true}

LibraryWebMobile.active=\texttt{true}

MicroStrategy Collaboration Server

- The state of whether MicroStrategy Collaboration Server is displayed in the MicroStrategy Installation Wizard:

CollaborationServer.visible=\texttt{Value}
The selection state of MicroStrategy Collaboration Server.

```
CollaborationServer.active=Value
```

For example, to select MicroStrategy Collaboration Server for installation, use:

```
CollaborationServer.visible=true
CollaborationServer.active=true
```

MicroStrategy Certificate Store

- The selection state of MicroStrategy Certificate Store.

```
CertificateManager.active=Value
```

- For example, to select MicroStrategy Certificate Store for installation, use:

```
CertificateManager.visible=true
CertificateManager.active=true
```

MicroStrategy Product and Component Installation Locations

You can define the installation locations for the following products and components:

- MicroStrategy Web Universal Install Location:

```
WebUniversal.deployDirectory=Value
```

- MicroStrategy Portlets Install Location:

```
Portlets.installDirectory=Value
```
- MicroStrategy GIS Connectors Install Location:
  
  `GISConnectors.installDirectory=Value`

- MicroStrategy Community Data Connectors
  
  `CommunityConnectors.installDirectory=Value`

- MicroStrategy Web Services J2EE Install Location:
  
  `WebServices.installDirectory=Value`

- MicroStrategy Mobile Server JSP Install Location:
  
  `MobileServer.installDirectory=Value`

- MicroStrategy Command Manager Install Location:
  
  `CommandManager.installDirectory=Value`

- MicroStrategy System Manager Install Location:
  
  `SystemManager.installDirectory=Value`

- Platform Analytics Install Location:
  
  `PlatformAnalytics.installDirectory=Value`

- MicroStrategy Identity Server Install Location:
  
  `UsherSecurityServer.installDirectory=Value`

- MicroStrategy Identity Manager Install Location:
  
  `UsherNetworkManager.installDirectory=Value`

- MicroStrategy Communicator Install Location:
  
  `UsherMobile.installDirectory=Value`
CPU License Information

This value should be specified when the license being used for MicroStrategy Intelligence Server is based on CPUs. Legal values are integers between 1 and either the number of CPUs allowed by the license or the number of CPUs in the machine, whichever is lower.

```
cpuCount.number=Value
```

By default, the maximum number of CPUs is allowed. This is represented with the following value for this parameter:

```
cpuCount.number=maximum
```

Agreement to Open Source Software Installation

This value is set to agree to the automatic download and installation of R. Set the value to `True` to have the installer download and install R. Set the value to `False` only if you have not selected the R Component for installation.

```
AgreeToDownloadOpenSourceSoftware=Value
```

Skip R Reinstall on Repair

R is reinstalled by default when performing a repair. Set this value to `True` if you do not want to reinstall R.

```
SkipReinstallR=Value
```

MicroStrategy Messaging Services Cluster Configuration

This increases the reliability, scalability and performance of Platform Analytics. To cluster Messaging Services set the property `MessagingServices.cluster.enable` to `True` and provide the hostname of at least 3 Messaging Services nodes. One in the
MessagingServices.cluster.localnode property and the rest in a comma separated list in MessagingServices.cluster.remotenodes.

- MessagingServices.cluster.enable=Value
- MessagingServices.cluster.localnode=Value
- MessagingServices.cluster.remotenodes=Value, Value

**MicroStrategy Identity Server Configuration: Tomcat Database**

MicroStrategy Identity Server installs a database that is a system of record for individual Usher identities. You can configure MicroStrategy Identity Server to communicate with the database, as part of the silent installation:

- The location of the Tomcat directory used by MicroStrategy Identity Server:

  UsherSecurityServer.tomcatDir=Path

  Provide a valid folder path that contains the correct version of Tomcat. You must be able to write to the webapps subfolder.

- The IP address for the machine that hosts the database:

  UsherSecurityServer.serverDBHost=IP_Address

- The port number for the database connection:

  UsherSecurityServer.serverDBPort=Port

- The account name for the database user that administers the database:
Installation and Configuration Guide

UsherSecurityServer.serverDBUser=User\Name

- The password for the database user specified above:
  UsherSecurityServer.serverDBPassword=User\Password

- The name of the MicroStrategy Identity Server database:
  UsherSecurityServer.serverDBInstance=DatabaseInstance

- The name of the database that stores log information for the MicroStrategy Identity Server:
  UsherSecurityServer.serverLogDBInstance=DatabaseInstance

MicroStrategy Identity Server Configuration: Ports and Certificates

Set up a trust relationship for MicroStrategy Identity Server using the Public Key Infrastructure (PKI), as part of the silent installation:

- The port used by MicroStrategy Identity Server for server (one-way SSL) authentication:
  UsherSecurityServer.serverPortOne=Port

- The port used by MicroStrategy Identity Server for client and server (two-way SSL) mutual authentication:
  UsherSecurityServer.serverPortTwo=Port

- The location of the public key SSL certificate file:
  UsherSecurityServer.serverSslCert=Path

- The location of the private key file:
  UsherSecurityServer.serverSslKey=Path
The location of the SSL certificate chain:

```
UsherSecurityServer.serverCaCert=Path
```

MicroStrategy Identity Server Configuration: Gateways

Set up a trust relationship for the Agent Gateway using the Public Key Infrastructure (PKI), as part of the silent installation:

- The port used by MicroStrategy Identity Server for the Agent Gateway (one-way SSL) authentication:

```
UsherSecurityServer.gatewayPort=Port
```

- The location of the public key SSL certificate file:

```
UsherSecurityServer.gatewaySslCert=Path
```

- The location of the private key file:

```
UsherSecurityServer.gatewaySslKey=Path
```

- The location of the SSL certificate chain:

```
UsherSecurityServer.gatewayCaCert=Path
```

MicroStrategy Identity Manager Configuration

You can configure MicroStrategy Identity Manager as part of the silent installation:

- The location of the Apache directory used by MicroStrategy Identity Manager:

```
UsherNetworkManager.apacheDir=Path
```
Provide a valid folder path that contains the conf and conf.d folders.

You must be able to write to the conf.d subfolder.

- The Apache user name:
  \[\text{UsherNetworkManager.apacheUser=UserName}\]

- The Apache group name:
  \[\text{UsherNetworkManager.apacheGroup=GroupName}\]

- Specify whether to use the same database connection as MicroStrategy Identity Server:
  \[\text{UsherNetworkManager.useSameDBSetting=false}\]

  By default, the same database connection is not used (set to false). If false is specified, you must define the database connection using the settings listed below.

- The IP address for the machine that hosts the database:
  \[\text{UsherNetworkManager.DBHost=IP_Address}\]

- The port number for the database connection:
  \[\text{UsherNetworkManager.DBPort=Port}\]

- The account name for the database user that administers the database:
  \[\text{UsherNetworkManager.DBUser=UserName}\]

- The password for the database user specified above:
  \[\text{UsherNetworkManager.DBPassword=UserPassword}\]
- The name of the MicroStrategy Identity Manager database:

  ```
  UsherNetworkManager.DBInstance=DatabaseInstance
  ```

- Specify whether any existing database instance should be dropped or preserved for new installations. This is set to `false` by default and preserves the schema and upgrades it if necessary. If this is set to `true`, drop the schema and create a fresh configuration.

  ```
  UsherNetworkManager.overwriteDb=Value
  ```

- Enable SSL for the MicroStrategy Identity Manager database connection. This is set to `false` by default. Valid values are `true` or `false`.

  ```
  UsherNetworkManager.DBenableSsl=Value
  ```

- Use this option to specify if you want to use the same SSL certificate as MicroStrategy Identity Server. If this value is set to `true`, use the same SSL certificates. If this is set to `false`, do not use the same SSL certificates.

  ```
  UsherNetworkManager.dbUseSameCert=Value
  ```

- Specifies the public key SSL certificate file (.crt)

  ```
  UsherNetworkManager.dbSslCert=Value
  ```

- Specifies the private key file (.key)

  ```
  UsherNetworkManager.dbSslKey=Value
  ```

- Specifies the Certification Authority's (CA) certificate file (.pem)

  ```
  UsherNetworkManager.dbCaCert=Value
  ```
MicroStrategy Library Web and Mobile

- Setting to automatically deploy Library Web and Mobile:
  
  ```
  LibraryWebMobile.deploy=Value
  ```

- Tomcat directory location for MicroStrategy Library Web & Mobile:
  
  ```
  LibraryWebMobile.tomcatDir=Value
  ```

MicroStrategy Platform Analytics

Connection Parameters to configure the database installed to store system telemetry:

- `PlatformAnalytics.DBHost=Value`
- `PlatformAnalytics.DBPort=Value`
- `PlatformAnalytics.DBUser=Value`
- `PlatformAnalytics.DBPassword=Value`
- `PlatformAnalytics.overwriteDb=Value`
- `PlatformAnalytics.upgrade.useDifferentDb=Value`

Topology Configuration

Workstation Topology helps administrators monitor and manage services in a MicroStrategy environment.

- Indicate a cluster environment. Define this parameter as `SINGLE` or `MULTIPLE` to indicate a single or multiple machine environment. The default value for this setting is `SINGLE`.

  ```
  ServicesRegistration.machine.environment=Value
  ```
• Provide all Services Registration server host names, separated by semicolon

```
ServicesRegistration.serverHosts=Value;Value
```

• Load balancer addresses:

Example: http://localhost/MicroStrategyMobile

```
WebServer.loadBalancer=Value
MobileServer.loadBalancer=Value
LibraryWebMobile.loadBalancer=Value
CollaborationServer.loadBalancer=Value
UsherServer.loadBalancer=Value
UsherNetworkManager.loadBalancer=Value
```

Registering MicroStrategy Processes as a Service

This setting registers MicroStrategy processes as a service to enable automatic start after system startup.

![Info](image)

Only root users may register processes as a service

```
RegisterServices=Value
```

Hide Select Components Dialog Box

Use this option to completely hide the product selection dialog during the execution of the MicroStrategy Installation Wizard. This option can be used in combination with individual product visibility/state options to have better control on the products to be installed/removed.
If the dialog is not visible, and no specific visibility/state options have been specified, default values will be used.

SelectComponents.visible=\text{Value}

Install on Demand Options

You can reduce the amount of data that has to be downloaded for the installation by excluding some of the .tzp files, located in the DataFiles folder, from the download. You can use this technique to download only the files required to complete your MicroStrategy installation, which can then also be used to reduce the amount of data packaged and downloaded for other MicroStrategy installations.

If you are performing a MicroStrategy Update 1 installation, you must include all of the files provided as part of the Update installation in their default location. This means that you cannot use the options below to point to the location of the Update installation files.

To reduce the amount of data required for MicroStrategy installations, you first need to determine the files required to support your installation of MicroStrategy, as described in Creating Custom Installation Packages, page 106. Once you determine and collect the .tzp files required to support your MicroStrategy installation, you can specify the location of these files using the following parameters:

- InstallOnDemand.style=\text{Value}

  Determines whether the required installation files are provided in a folder or at a URL. You must define this parameter with one of the following values:
  - FileSystem: Type this value if the required installation files are stored in a folder on the local machine or a server machine. You must also provide the location of the files using the InstallOnDemand.sourceLocation parameter.
• **HTTP**: Type this value if the required installation files are stored at an unsecured URL. You must also provide the location of the files using the `InstallOnDemand.url` parameter.

• **HTTPS**: Type this value if the required installation files are stored at a secured URL. You must also provide the location of the files using the `InstallOnDemand.url` parameter, as well as the user name and password to access the URL using the `InstallOnDemand.username` and `InstallOnDemand.password` parameters.

• `InstallOnDemand.sourceLocation=Value` Location of the folder that stores any required installation files. Type the location of the local file path. If you store the files in a local folder, do not provide a location for the `InstallOnDemand.url` parameter.

• `InstallOnDemand.url=Value` Location of the URL for the HTTP or HTTPS location that stores any required installation files. Type the URL for the location that stores any required installation files. If you store the files at an HTTP or HTTPS location, do not provide a location for the `InstallOnDemand.sourceLocation` parameter.

• `InstallOnDemand.bypassCertificateChecking=Value` If you retrieve the installation files from a URL location using HTTPS, you can use this setting to skip any certificate checking by defining this option to `true`. To maintain certificate checking, define this option as `false`.

• `InstallOnDemand.username=Value` If you retrieve the installation files from a URL location, type a user name that has access to the URL location. If there is no login required to the URL or you retrieve the installation files from a local folder, you do not need to define a value for this parameter.

• `InstallOnDemand.password=Value` If you retrieve the installation files from a URL location, type a password for the user name. If there is no login required to the URL or you retrieve the installation files from a local folder, you do not need to define a value for this parameter.
Unique Post-Installation Configurations

MicroStrategy supports many different Linux environments with various system configurations. There are a few cases in which you must perform some manual configurations to support the use of MicroStrategy on your system. For more information, see *Unique Post-Installation Configurations, page 193* in *Chapter 3, Installing MicroStrategy on Linux*.

Silent Installation Output

The installation returns 0 if the installation is successful, and any other value if it is not. The *install.log* file in the *InstallPath* directory provides more information on possible errors. For more information on the *install.log* file, see *Installation Log File, page 556* in *Chapter 2, Installing MicroStrategy on Windows*.

If the installation fails on any of the steps before it starts copying the files, it does not give any feedback other than the return value different from 0.

Activating a Silent Installation

After the silent installation is complete, you must activate the MicroStrategy installation within 30 days. To activate the installation you can follow the instructions provided in *Chapter 5, Activating Your Installation*.

Configuring MicroStrategy in Command Line Mode

The MicroStrategy Configuration Wizard is provided in command line mode so that you can use the Configuration Wizard through the operating system console if you do not have access to the GUI. Running the Configuration Wizard in command line mode to configure

Configuring your MicroStrategy Installation

After completing the steps to install MicroStrategy products, you can set up and configure your installation. To help guide the rest of your installation and configuration steps, see Installing and Configuring MicroStrategy on Linux, page 112 in Chapter 1, Planning Your Installation, for an installation and configuration checklist.
DEPLOYING OEM APPLICATIONS
This chapter explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.

The MicroStrategy platform can be deployed as an OEM application in various ways:

- MicroStrategy can be deployed as a software as a service model through the use of MicroStrategy Web. In this scenario, MicroStrategy is installed and configured at a centralized location using the standard process, and the customized application is deployed as an OEM application using MicroStrategy Web. For information on deploying MicroStrategy Web, see Chapter 7, *Deploying MicroStrategy Web and Mobile Server*. For information on customizing MicroStrategy Web, see *Customizing MicroStrategy Web, page 686*.

- MicroStrategy can be deployed as part of an OEM software bundle directly to a customer environment. This chapter focuses on the development and deployment of this type of OEM application.

The following is a list of best practices on how to deploy MicroStrategy as an OEM application. Additionally, you can use MicroStrategy System Manager. System Manager allows you to define multiple configurations for your MicroStrategy environment that can be executed in a single workflow. For steps to use System Manager to deploy MicroStrategy configurations, see the *System Administration Guide*.

- Install MicroStrategy on an OEM environment. For installation information, see:
  - *Chapter 2, Installing MicroStrategy on Windows*
  - *Chapter 3, Installing MicroStrategy on Linux*
Create data source connections using the Connectivity Wizard, as described in *Creating DSNs for OEM Environments, page 684.*

Configure MicroStrategy Intelligence Server using the Configuration Wizard. This tool allows you to save configurations as response files that can be used to automate the configuration for the OEM deployment. This allows you to re-use all the configurations performed when developing an embedded application for the deployment process as well, as described in *Configuring a MicroStrategy Installation, page 685.*

Design projects and a reporting environment. You can use the various MicroStrategy products and relevant documentation to create the required MicroStrategy environment. For additional best practices when designing a reporting environment, see *Designing a Project and Reporting Environment, page 686.*

Customize MicroStrategy Web through the use of the MicroStrategy SDK, as described in *Customizing MicroStrategy Web, page 686.*

Deploy a MicroStrategy OEM application on an OEM's customer environment, as described in *Deploying a MicroStrategy OEM Application, page 687.*

Create data source connections on your customer environment as necessary, as described in *Creating DSNs for OEM Environments, page 684.*

Configure and tune an embedded deployment through the use of various MicroStrategy tools, as described in *Tuning an OEM Deployment, page 694.*

If you are modifying a project that has already been deployed as an OEM application, see *Updating OEM Applications, page 695* for best practices on how to incorporate any custom reports or objects that may have been created for the deployed application.
Creating DSNs for OEM Environments

Establishing communication between MicroStrategy and your databases or other data sources is an essential first step in configuring MicroStrategy products for reporting and analysis of your data. These data sources are used to store the data warehouse and the MicroStrategy metadata, which are both required to support a MicroStrategy reporting environment.

To create a connection to these data sources you need an ODBC driver as well as a data source name (DSN). MicroStrategy comes packaged with ODBC drivers to support connecting to various data sources. For more information on ODBC drivers, see Communicating with Databases, page 356.

When setting up your OEM environment, you must create a separate DSN to connect to the main data source and the metadata repository. This requirement is true even if the data source and metadata repository are stored in the same database or other data source. The main data source and the metadata are described below:

- A data source stores the data that users of the system can analyze through BI capabilities offered by MicroStrategy products.

- Metadata is a repository whose data associates the tables and columns of a data warehouse with user-defined attributes and facts to enable the mapping of business views, terms, and needs to the underlying database structure. Metadata can reside on the same server as the data warehouse or on a different server. It can be also be stored in a different relational database than your data warehouse. A metadata can be created using the Configuration Wizard, as described in Configuring a MicroStrategy Installation, page 685.

A DSN can be created using the MicroStrategy Connectivity Wizard, as described in Defining DSNs, page 362.
Creating DSNs as Part of an OEM Deployment

As part of the deployment of an OEM application, the Connectivity Wizard can run unattended to create DSNs on Linux environments. This allows you to perform this configuration automatically using scripts. For information on creating ODBC DSNs using the command line version of the Connectivity Wizard, see Creating a DSN for a Data Source, page 699.

For OEM deployments on Windows machines, use the Connectivity Wizard interface to create ODBC DSNs, as described in Defining DSNs, page 362.

Configuring a MicroStrategy Installation

After installing MicroStrategy, you can use the MicroStrategy Configuration Wizard to configure the metadata repository, statistics tables, History List tables, MicroStrategy Intelligence Server, and multiple project sources.

The Configuration Wizard interface guides you through each of these configurations, as described in Initial MicroStrategy configuration, page 370.

In addition, all configurations that are performed using the Configuration Wizard can be saved as response files. These files can then be used later to automate much of the initial configuration of MicroStrategy when deploying it as an OEM application. This allows you to re-use all the configurations performed when developing an OEM application. For information on configuring MicroStrategy using a response file, see Configuring MicroStrategy with a Response File, page 404.
Designing a Project and Reporting Environment

You can use the various MicroStrategy products and relevant documentation to create the required MicroStrategy reporting environment for your OEM application. The following best practices can be helpful when creating this reporting environment:

- It is common to define objects such as reports, documents, attributes, metrics, and filters that are created for the OEM application so that they cannot be modified once it is deployed. You can modify the object security of each object so that it does not allow write access. This ensures that the reports provided out of the box with the OEM application are not modified and overwritten. Users can still use Save As to save their own personal copies of any objects to make any required changes.

- You can modify the folder permissions in MicroStrategy to determine where reports and objects can be created. Limiting the folders that allow write access can require users to create reports in their My Reports folder.

- If you are modifying a project that has already been deployed as an OEM application, see *Updating OEM Applications, page 695* for best practices on how to incorporate any custom reports or objects that may have been created for the deployed application.

Customizing MicroStrategy Web

MicroStrategy Web provides users with a highly interactive environment and a low-maintenance interface for reporting and analysis. Using the MicroStrategy Web interface, users can access, analyze, and share corporate data through any web browser on any operating system.
With the MicroStrategy SDK, you can customize, embed, or extend MicroStrategy Web into your application, or modify the standard interface or functionality. Common customizations for OEM deployments include:

- Customizing the look and feel of the MicroStrategy Web interface. This can include changing the color scheme, adding or removing content, using customized logos, and many other customizations.

- Integrating MicroStrategy Web with third-party applications such as:
  - Portals
  - External security and user management systems
  - Advanced data visualizations

- Extending functionality to support composite applications includes such things as writeback capabilities and other custom features.

To customize MicroStrategy Web using the MicroStrategy SDK, refer to the MicroStrategy Developer Library (MSDL). The MSDL contains details about the architecture, object models, customization scenarios, code samples, and so on that are useful for building a sophisticated and highly functional, customized application.


**Deploying a MicroStrategy OEM Application**

After an OEM application is developed, it then must be deployed to the customer's environment. Steps to deploy an embedded
installation include:

1. Installing the required MicroStrategy products on the customer’s environment. This can be automated using silent installation techniques, as described in OEM Silent Installations, page 688. To use the Installation Wizard to install MicroStrategy products, see:
   - Chapter 2, Installing MicroStrategy on Windows
   - Chapter 3, Installing MicroStrategy on Linux

2. Additional configurations are required, as described in Configuring an OEM deployment installation, page 693.

OEM Silent Installations

You can use silent installation to easily embed MicroStrategy products with other applications. The steps below show you how to use a silent installation to deploy an OEM application on a Windows environment. For additional information on silent installations, see Chapter 9, Automated Installation on Windows.

You can follow the steps below to perform a silent installation on a Windows environment. To perform a silent installation on a Linux environment, see Silent Installation, page 625.

Ensure that the MicroStrategy installation files are accessible on the machine in which the installation is being performed. If a required installation file is not accessible, the installation can fail, often providing a warning about missing requirements.

To Perform an OEM Silent Installation

When MicroStrategy products are installed as software bundled with another product, the following procedure is strongly recommended:
1. Create an installation response file (response.ini) for the MicroStrategy products to install. The table that follows shows which sections of the file are mandatory and which are optional.

For detailed information regarding the contents of the response.ini file, see Configuring a response.ini file to Install MicroStrategy, page 558.

<table>
<thead>
<tr>
<th>Response File Section</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Installer]</td>
<td>Required</td>
</tr>
<tr>
<td>HideAllDialogs =</td>
<td>Required</td>
</tr>
<tr>
<td>PreventReboot =</td>
<td>Optional</td>
</tr>
<tr>
<td>StopAllServices =</td>
<td>Optional</td>
</tr>
<tr>
<td>StopIIS =</td>
<td>Optional</td>
</tr>
<tr>
<td>CheckRenameOperations =</td>
<td>Optional</td>
</tr>
<tr>
<td>[UserRegistration]</td>
<td>Required</td>
</tr>
<tr>
<td>[ComponentSelection]</td>
<td>Required</td>
</tr>
<tr>
<td>EnterpriseManagerSelect =</td>
<td>Required</td>
</tr>
<tr>
<td>[InitialPaths]</td>
<td>Required</td>
</tr>
<tr>
<td>EnterpriseManager =</td>
<td>Required</td>
</tr>
</tbody>
</table>

Setting HideAllDialogs = TRUE causes the script for the response file to:

- Use default values as specified in the response.ini file.
- Not require user input.
- Keep the dialog flow consistent from one instance to the next.

Consistency in the response file script from one instance to the
next is necessary; if setup.iss detects an inconsistency in
the dialog flow, installation is terminated and a log file for the
failure is created.

The only dialog flow modifications pertinent to silent installation
are specific to file location. Therefore, the only portion of the
response.ini that may need to be modified is the
[InitialPaths] section.

The rest of this procedure assumes you have saved the
response.ini file to the file path C:\. If you save it to another
file path, replace C:\response.ini with the file path of your
response.ini file.

You must save the response.ini file as ANSI encoding.

2. Create the setup.iss file to use in conjunction with the
response.ini file for the silent installation. Use a text editor to
create the setup.iss file with the following information:

   [InstallShield Silent]
   Version=v7.00
   File=ResponseFile
   [File Transfer]
   OverwrittenReadOnly=NoToAll
   [Application]
   Name=MicroStrategy
   Version=x.y.z
   #x.y.z represent the version of the MicroStrategy platform
   Company=MicroStrategy
   Lang=LanguageValue
   [8CCF3F6C-55B7-4A27-8C68-ADF21D0585A2]-DlgOrder]
   Count=0

You must save the setup.iss file as ANSI encoding.

The version in the setup.iss file must match the MicroStrategy
version you are installing exactly. For example, if you are installing
version 10.7.0 you must enter `Version=10.7.0`. Entering a version as `Version=10` or `Version=10.7.x` causes an error when trying to perform a silent installation of version 9.3.0.

For an explanation of the `LanguageValue` parameter within the line `Lang=LanguageValue`, see *Language Settings for Silent Installations, page 692*.

3. From the Windows Start menu, select **Programs**, then select **Accessories**, and then right-click **Command Prompt** and select **Run as Administrator**. The User Account Control dialog box opens.

   The steps to open a Windows command prompt with administrator privileges may be different depending on your version of Windows.

4. Click **Yes** to open the command prompt with administrator privileges. The command prompt is displayed.

5. Run the silent install with the `response.ini` file in conjunction with the `setup.iss` file as follows:

   ```
   INSTALL_PATH\setup.exe -LLanguageValue --ResponseFile="C:\response.ini" -s -f1"c:\setup.iss" -f2"c:\setup.log"
   ```

   In the syntax shown above, the `-s` parameter indicates that the installation is to be completely silent. If the `-s` parameter is not included in the command, then an interface is displayed during the installation that shows the progress of the installation.
If the setup program encounters an invalid value for an installation requirement, the setup terminates and the silent installation is ended. You can review any errors in the setup.log file.

6. If a restart is required after the installation is complete, a restart of the machine is automatically triggered. Power the machine back on to allow for the completion of any configurations that are required after the restart of the machine.

7. After the installation is complete, you can check the result of the installation process. If the silent installation is successful, the resulting code value is zero (ResultCode=0) in the setup.log file. This is the only indication of the installation being completed if the installation is completely silent and a restart of the machine is not required.

Language Settings for Silent Installations

In the final steps of the procedure to run an OEM silent installation:

- You can set MicroStrategy Developer language settings by setting the language value in the setup.iss file.

- You can bypass the language prompt by running setup.exe with the command line option for the language.

The following table lists the values for the different languages that MicroStrategy supports:

<table>
<thead>
<tr>
<th>Language</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danish</td>
<td>0006</td>
</tr>
<tr>
<td>Dutch</td>
<td>0019</td>
</tr>
<tr>
<td>English</td>
<td>0009</td>
</tr>
</tbody>
</table>
For example, to select English as the language:

- **For the setup.iss file**, change `Lang=LanguageValue` to:
  
  `Lang=0009`

- **To run the silent install**, use the command line option as follows:

  `Path\setup.exe -L0009`

  For the command line option, you must type `-L` in front of the language code to signify that you are entering a language.

### Configuring an OEM deployment installation

Once MicroStrategy has been installed on the customer's environment, the following additional steps must be taken to prepare the initial configuration of MicroStrategy software:
• Provide the MicroStrategy metadata that was developed for the managed application.

• Create separate DSNs to connect to the data warehouse and the metadata, as described in Creating DSNs for OEM Environments, page 684.

• Use the Configuration Wizard to configure metadata, Intelligence Server, and project sources. If you saved your configurations as response files, these configurations can be re-used for automated configuration. For information on using the Configuration Wizard, see Configuring a MicroStrategy Installation, page 685.

• Configure and tune an OEM deployment through the use of various MicroStrategy tools, as described in Tuning an OEM Deployment, page 694.

Tuning an OEM Deployment

A MicroStrategy OEM deployment requires additional tuning and configuration, both during deployment and throughout the life cycle of the OEM application. Various ways to perform these configurations are described below:

Tuning with Command Manager

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 Developer or Narrowcast Administrator interface.
Developers of 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.

OEM application deployments typically required on-premises configuration of environment-specific settings such as database user and password, governing options, caching options, and other tuning requirements. Command Manager Runtime scripts enable OEMs to automate a number of such configuration settings.

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, refer to the [System Administration Guide].

**Configuring MicroStrategy in Command Line Mode**

The MicroStrategy Configuration Wizard is provided in command line mode so that you can use the Configuration Wizard through the operating system console if you do not have access to the GUI. Running the Configuration Wizard in command line mode to configure MicroStrategy on Linux machines is covered in Configuring MicroStrategy with a response.ini file, page 703.

**Updating OEM Applications**

The lifecycle of an OEM application often requires the OEM application to be updated with new reports and other enhancements.
These enhancements can be developed within the OEM application and then the customer's existing application can be updated.

For more information on updating OEM applications, see:

**Modifying Deployed OEM Applications**

If you are modifying a project that has already been deployed as an OEM application, you must update the application in a way that does not disrupt any current customer development, as described below:

- Retrieve the customer's metadata so that custom reports or other objects that have been created can also be included in the OEM application update.

- Any new objects deployed as part of the OEM application update should be tested to ensure that they do not negatively affect the objects provided with the previous deployment of the OEM application. MicroStrategy Integrity Manager can be used to automate the testing of including new objects in an OEM application. For information on Integrity Manager, refer to the System Administration Guide.

**Deploying an OEM Application Update**

Once updates for an OEM application are complete, the application changes must then be deployed. There are two ways in which the update can be deployed, as described below.

**Replacing the entire MicroStrategy project in the existing system**

Once updates for an OEM application are complete, the project can be replaced in an OEM application. This is commonly done by duplicating the updated OEM project, and then merging that project into the production OEM application.
These tasks can be achieved using the MicroStrategy Project Duplication Wizard and Project Merge Wizard. Both of these tools can perform their tasks from the command line, which can allow the project duplication and replacement process to be automated. The steps to use these tools to duplicate and replace a project are provided in the System Administration Guide.

Merging modified objects into a project in an OEM application

Once updates for an OEM application are complete, the updates to the project can be deployed through the use of 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 conflicts involving the objects are resolved.

These update packages can be developed using Object Manager. Once the package is ready for deployment, it can be deployed using the graphical interface with Object Manager or using the unattended execution with Command Manager Runtime. For information on creating and deploying an update package with Object Manager, refer to the information on managing projects provided in the System Administration Guide. For information on Command Manager Runtime, refer to the System Administration Guide.
CONFIGURING MICROSTRATEGY USING COMMAND LINE TOOLS
MicroStrategy tools are provided in command line mode on Linux so that you can perform various configuration tasks through the operating system console. This enables you to perform your required configurations even if you do not have access to the MicroStrategy interface.

On a Windows machine, it is recommended to use the appropriate MicroStrategy interfaces to perform the configurations described in this chapter.

When you perform MicroStrategy configuration tasks through a Linux operating system console, you must make sure reserved words and characters are not mistakenly included in your commands. To avoid issues caused by reserved words and characters, see Supporting Reserved Words and Characters, page 733.

This section covers the configurations listed below:

Creating a DSN for a Data Source

After you install an ODBC driver (see Appendix A, Connecting to Databases and Data Sources), you can define one or more data sources for it. The DSN should provide a unique description of the data, for example, Payroll_Project_Metadata or Payroll_Warehouse.

The DSN is the name for a pointer used by a client application (in this case MicroStrategy) to find and connect to a data source. Multiple DSNs can point to the same data source and one DSN can be used by different applications.

MicroStrategy provides a one-line command line version of the MicroStrategy Connectivity Wizard to create DSNs on Linux.
You can create DSNs using the MicroStrategy Connectivity Wizard on Windows and Linux machines, as described in *Communicating with Databases, page 356.*

To create a DSN on Linux from the command line

1. From a Linux console window, browse to \HOME\_PATH, where \HOME\_PATH is the directory that you specified as the home directory during installation.

2. Browse to the folder *bin.*

3. Type `mstrconnectwiz -h` and then click *Enter* to display command line syntax and examples for different database platforms.

4. Create your command based on the syntax and examples displayed. For example, the command below creates a DSN for an Oracle database and tests login credentials:

   ```
   mstrconnectwiz ORCLW MyOracleDSN 12.34.56.78 orcl 1521 -u:OracleUser -p:OracleUserPassword
   ```

Testing ODBC connectivity

ODBC connectivity is one of two layers of connectivity that are listed in the next table, along with the associated connectivity testing programs. Connectivity should be tested from the bottom up—the network layer first and then the ODBC layer.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Test with</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODBC driver</td>
<td>Test ODBC</td>
</tr>
<tr>
<td></td>
<td><code>mstrtestodbc</code> or <code>mstrtodbcx</code></td>
</tr>
<tr>
<td>Network</td>
<td>Simple Network Layer Testing Tool</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Ping, <code>PING.EXE</code>, for TCP/IP</td>
</tr>
</tbody>
</table>
The test method described above reflects the situation when the ODBC driver and the database network software are bundled. If they are not bundled, they must be configured and tested separately, using database-specific tools.

Using the DB Query Tool

The MicroStrategy DB Query Tool is available in Windows, and Linux Intelligence Server installations. It is used to test and troubleshoot connectivity to databases, create and execute SQL commands through ODBC, and run scripts.

Before you use the DB Query Tool, test the network layer with the network layer utility, PING.EXE. Consult your operating system or network system documentation for details.

To use the DB Query Tool

1. To use the DB Query Tool:

   - On Windows using the DB Query Tool interface, perform the following step:
     
     From the Windows Start menu, go to Programs > MicroStrategy Tools > DB Query Tool.

   - On Windows from the command line, perform the following steps:
     
     1. From the Windows Start menu, select Run.
     2. In the Open drop-down list, type cmd and click OK. A command prompt opens.
     3. Type todbcx.exe and click Enter. Prompts guide you through testing your ODBC connection from the command
In allation and Configuration Guide

line and should be used in place of the steps below. For detailed steps on how to use the command line version of this tool, see Testing ODBC connectivity in Chapter 12, Configuring MicroStrategy Using Command Line Tools.

- On Linux using the DB Query Tool interface, perform the following steps:

  1. In a Linux console window, browse to HOME_PATH, where HOME_PATH is the directory that you specified as the home directory during installation.

  2. Browse to the folder bin and type ./mstrdbquerytool, then click Enter.

- On Linux from the command line, perform the following steps:

  1. In a Linux console window, browse to HOME_PATH, where HOME_PATH is the directory that you specified as the home directory during installation.

  2. Browse to the folder bin and type ./mstrtodbcx, then click Enter. Prompts guide you through testing your ODBC connection from the command line and should be used in place of the steps below. For detailed steps on how to use the command line version of this tool, see Testing ODBC connectivity in Chapter 12, Configuring MicroStrategy Using Command Line Tools.

  2. From the Session menu, select Open Connection, or click the Connect icon on the toolbar. The Connect dialog box opens. The connection interface varies depending on the destination database.

  3. Select the DSN for a data source.

  4. Enter the appropriate user name and password.
5. Click **Connect**. After your connection is opened, the connection string is displayed in the MicroStrategy DB Query Tool at the bottom. Your cursor is inserted automatically in the SQL Statement window.

6. In the SQL Statement window, type a SQL query such as:

```sql
select count (*) from Table

where Table is a system-defined table, such as SYSOBJECTS for Microsoft SQL Server or a MicroStrategy-created table such as DSSMDSYSPROP in the MicroStrategy metadata.
```

7. From the **Queries** menu, select **Execute Query**. A table of data from the database is displayed in the Query Result window.

8. From the **Session** menu, select **Close Connection** to close the database connection.

9. From the **File** menu, select **Exit**.

**Configuring MicroStrategy with a response.ini file**

The MicroStrategy Configuration Wizard is provided in command line mode so that you can use the Configuration Wizard even if you do not have access to a GUI. You can perform the following configurations with the Configuration Wizard in command line mode:

- Create metadata, statistics, and History List tables
- Create new MicroStrategy Intelligence Server definitions in the metadata, assign an existing server definition for Intelligence Server, and delete existing server definitions
- Create MicroStrategy project sources in a server (three-tier) mode
Direct (two-tier) data sources are available only on the Windows operating system.

Using the Configuration Wizard in command line mode creates a response.ini file. This file can then be used from the command line to configure MicroStrategy without stepping through the pages of the Configuration Wizard. You can also distribute a response.ini file to other users and machines to perform multiple configurations without stepping through the Configuration Wizard for each configuration.

This section covers the following procedures and information related to configuring MicroStrategy from the command line on a Linux machine using a response.ini file:

- Creating a response.ini File, page 704
- Using the response.ini file to Configure MicroStrategy, page 727
- Parameters and Options in the response.ini File, page 727

You can also configure MicroStrategy using the Configuration Wizard in command line mode on a Windows machine. However, on a Windows machine, it is recommended to use the Configuration Wizard graphical user interface to create and use a response file, which is described in Configuring MicroStrategy with a Response File, page 404.

Before you can configure MicroStrategy with the Configuration Wizard in command line mode, you must ensure that you meet the prerequisites listed in Configuration Wizard Prerequisites, page 370.

Creating a response.ini File

This section describes how to configure MicroStrategy using the command line mode. Performing the steps in this section creates a response.ini file that can be used to configure MicroStrategy installations on Linux machines.
To configure MicroStrategy using the command line mode

1. In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation.

2. Browse to the bin directory.

3. At the command prompt, type mstrcfgwiz-editor, then click Enter. The Configuration Wizard opens in command line mode.
   
   The sections or pages of the wizard displayed depend on your selections.

4. Click Enter.

5. You can select to use a response.ini file to configure MicroStrategy, or create a new response.ini file to support a new configuration, as described below:

   - Type 1, and then click Enter to use a response.ini file to configure MicroStrategy. For steps to use a response.ini file in command line mode, see Using the response.ini file to Configure MicroStrategy, page 727.

   - Type 2, and then click Enter to create a new response.ini file.

6. You can support the configuration tasks described in the sections listed below:

   - Type 1, and then click Enter to create metadata, History List, and statistics tables. Refer to Creating metadata, History List, and statistics tables, page 706 for steps to create metadata and statistics tables.

   - Type 2, and then click Enter to configure a MicroStrategy Intelligence Server definition. Refer to Setting up MicroStrategy Intelligence Server, page 715 for steps to configure an Intelligence Server definition.
• Type 3, and then click **Enter** to create project sources. Refer to *Creating a project source, page 721* for steps to create project sources.

Creating metadata, History List, and statistics tables

If you selected option 1 in *Creating a response.ini File, page 704*, you can create metadata tables, History List tables, and statistics tables.

Creating metadata tables

You can create metadata tables in a data source, as described in the procedure below.

If metadata tables already exist in the location you plan to store your metadata tables in and you do not want to overwrite the current metadata tables, you should use the option described below.

This procedure assumes you have already opened the Configuration Wizard in command line mode and selected to create metadata and statistics tables, as described in *Creating a response.ini File, page 704*.

---

**To create metadata tables**

1. In the prompt asking whether to create metadata tables, type **Y**, and then click **Enter**. You are then prompted for ODBC data source information.

2. Type the number corresponding to the ODBC DSN for the database to store your metadata tables, and then click **Enter**.

   If you do not have a DSN defined on your Linux machine, see *Creating a DSN for a Data Source, page 699*.
3. If the Configuration Wizard detects an existing metadata repository in the database location you specified, a message is displayed on whether to re-create the metadata tables. If you type \texttt{Y} and press enter, all information in the existing metadata repository is overwritten when the response file is executed at a later time.

4. Enter a login and password to your DSN and click \texttt{Enter}.

5. Enter characters to use as a prefix for the names of your metadata tables or use no prefix and click \texttt{Enter}. The next steps depend on your ODBC data source details.

To configure DB2 MVS database options

6. Enter the database name to use or use the default name and click \texttt{Enter}.

7. Enter characters to use as a table space name for your metadata tables or use the default table space name and click \texttt{Enter}.

To select a metadata script

8. Select the script used to create the metadata tables or use the default script and click \texttt{Enter}

9. Type \texttt{Y}, and then click \texttt{Enter} to create History List tables.

10. Type \texttt{N}, and then click \texttt{Enter} to skip History List table creation.

11. Type \texttt{Y}, and then click \texttt{Enter} to create statistics tables.

12. Type \texttt{N}, and then click \texttt{Enter} to skip statistics tables creation.

13. By default, the configuration is saved as \texttt{Response.ini} in the \\
\texttt{/HOME\_PATH/} directory, where \texttt{HOME\_PATH} is the directory you specified as the Home Directory during installation. You can leave the field blank to use the default name or type a different
name, and then click Enter. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

14. Type Y, and then click Enter.

15. Type N, and then click Enter to quit without running the configuration.

Creating History List tables

You can create History List tables in a data source, as described in the procedure below.

This procedure assumes you have already opened the Configuration Wizard in command line mode and selected to create metadata, History List, and statistics tables, as described in Creating a response.ini File, page 704.

To create History List tables

1. After you create metadata tables (see Creating metadata tables, page 706) or skip both of the creation of metadata tables, you are prompted to create History List tables.

   To create History List tables, type Y, and then click Enter.

2. You can supply ODBC DSN information in various ways described below, which depend on whether you previously created metadata tables as part of the configuration process:

   - If you did not create metadata tables as part of the configuration process, you are prompted to enter ODBC DSN information. The steps to enter this information is described in the To provide ODBC DSN information, page 709 section within this procedure.
• If you created metadata tables as part of the configuration process, you are prompted whether to use the same metadata table ODBC DSN information for your History List tables. You have the following options:

• Type `Y`, and then click **Enter** to create History List tables with the same ODBC DSN information entered for your metadata tables. You are then prompted to select a History List script to create the History List tables, which is described in the *To select a History List script, page 710* section within this procedure.

• Type `N`, and then click **Enter** to provide different ODBC DSN information, which is described in the *To provide ODBC DSN information, page 709* section within this procedure.

**To provide ODBC DSN information**

1. Type the number corresponding to the ODBC DSN for a database to create your History List tables in, and then click **Enter**.

   If you do not have a DSN defined on your Linux machine, see *Creating a DSN for a Data Source, page 699*.

2. Depending on your database type, you may be prompted to provide a login and password to your DSN:

   • Type a login name for your database to create your History List tables in, and then click **Enter**.

   • Type a password for the login name provided, and then click **Enter**.

3. The next configuration displayed depends on your ODBC data source details:

   • If the data source points to a DB2 MVS database, steps to configure a DB2 MVS database are displayed. These are
To configure DB2 MVS database options

1. You can enter the database name to use or use the default name, as described below:
   - Type the database name to use, and then click Enter.
   - Leave the prompt blank, and then click Enter to use the default database.

   You are then prompted to provide the MVS table space name.

2. You can enter characters to use as a table space name for your metadata tables or use the default table space name, as described below:
   - Type the required table space name characters, and then click Enter.
   - Leave the prompt blank, and then click Enter to use the default table space name.

   You are then prompted to select a statistics script to create statistics tables.

To select a History List script

You can select the script used to create the History List tables or use the default script, as described below:
- Enter a valid path to a script file, and then click **Enter**.
- Leave the field blank, and then click **Enter** to use the default script for your database type.

To create statistics tables

- Type **Y**, and then click **Enter** to create statistics tables. Creating statistics tables is described in *Creating statistics tables, page 712.*
- Type **N**, and then click **Enter** to skip statistics tables creation. You are then prompted to provide a name for the response.ini file, as described in the *To generate a response.ini file, page 711* section within this procedure.

To generate a response.ini file

1. By default, the configuration is saved as **Response.ini** in the `/HOME_PATH/` directory, where `HOME_PATH` is the directory you specified as the Home Directory during installation. You can leave the field blank to use the default name or type a different name, and then click **Enter**. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

2. You can choose to run the configuration you just completed or to run the configuration using the response.ini file at a different time, as described below:
   - Type **Y**, and then click **Enter** to run the configuration.
   - You can also use the response.ini file created for future configurations, as described in *Using the response.ini file to Configure MicroStrategy, page 727.*
   - Type **N**, and then click **Enter** to quit without running the configuration. You can use the response.ini file created for...
future configurations, as described in *Using the response.ini file to Configure MicroStrategy, page 727.*

Creating statistics tables

This procedure assumes you have already opened the Configuration Wizard in command line mode and selected to create metadata, History List, and statistics tables, as described in *Creating a response.ini File, page 704.*

To create statistics tables

1. After you create metadata tables (see *Creating metadata tables, page 706*), create History List tables (*Creating History List tables, page 708*), or skip both of these procedures, you are prompted to create statistics tables.

   To create statistics tables, type *Y*, and then click **Enter**.

2. You can supply ODBC DSN information in various ways described below, which depend on whether you previously created metadata tables as part of the configuration process:

   - If you did not create metadata tables as part of the configuration process, you are prompted to enter ODBC DSN information. The steps to enter this information is described in the *To provide ODBC DSN information, page 713* section within this procedure.

   - If you created metadata tables as part of the configuration process, you are prompted whether to use the same metadata table ODBC DSN information for your statistics tables. You have the following options:

     - Type *Y*, and then click **Enter** to create statistics tables with the same ODBC DSN information entered for your metadata
tables. You are then prompted to select a statistics script to create the statistics tables, which is described in the *To select a statistics script, page 714* section within this procedure.

- Type N, and then click **Enter** to provide different ODBC DSN information, which is described in the *To provide ODBC DSN information, page 713* section within this procedure.

**To provide ODBC DSN information**

1. Type a valid ODBC DSN for a database to create your statistics tables in, and then click **Enter**. You are then prompted to provide a login to your DSN.

2. Type a login name for your database to create your statistics tables in, and then click **Enter**. You are then prompted to provide a password for the login name.

3. Type a password for the login name provided, and then click **Enter**.

The next configuration displayed depends on your ODBC data source details:

- If the data source points to a DB2 MVS database, steps to configure a DB2 MVS database are displayed. These are described in the *To configure DB2 MVS database options, page 714* section within this procedure.

- If the data source does not point to a DB2 MVS database, the step to select a statistics script to create statistics tables is displayed. This step is described in the *To select a statistics script, page 714* section within this procedure.
To configure DB2 MVS database options

1. You can enter the database name to use or use the default name, as described below:
   - Type the database name to use, and then click **Enter**.
   - Leave the prompt blank, and then click **Enter** to use the default database.
   
   You are then prompted to provide the MVS table space name.

2. You can enter characters to use as a table space name for your metadata tables or use the default table space name, as described below:
   - Type the required table space name characters, and then click **Enter**.
   - Leave the prompt blank, and then click **Enter** to use the default table space name.
   
   You are then prompted to select a statistics script to create statistics tables.

To select a statistics script

You can select the script used to create the statistics tables or use the default script, as described below:

   - Enter a valid path to a script file, and then click **Enter**.
   - Leave the field blank, and then click **Enter** to use the default script for your database type.

   You are then prompted to provide a name for the **response.ini** file.
To generate a response.ini file

1. By default, the configuration is saved as Response.ini in the
   /HOME_PATH/ directory, where HOME_PATH is the directory you
   specified as the Home Directory during installation. You can
   leave the field blank to use the default name or type a different
   name, and then click Enter. The response.ini file is
   generated, and you are prompted whether to run the
   configuration immediately.

2. You can choose to run the configuration you just completed or to
   run the configuration using the response.ini file at a different
   time, as described below:
   - Type Y, and then click Enter to run the configuration.
   - You can also use the response.ini file created for future
     configurations, as described in Using the response.ini file to
   - Type N, and then click Enter to quit without running the
     configuration. You can use the response.ini file created for
     future configurations, as described in Using the response.ini

Setting up MicroStrategy Intelligence Server

If you selected option 2 in Creating a response.ini File, page 704, you
can set up MicroStrategy Intelligence Server to create, use, or delete
server definitions. To begin setting up your server definition, you must
enter information about your ODBC DSN and MicroStrategy
connections.
To set up MicroStrategy Intelligence Server

1. Type the corresponding number for the ODBC DSN for a database to connect Intelligence Server to. This should be the data source that stores your metadata. Then click Enter. You are then prompted to provide a login to your DSN.

2. Type a login name for your database to create your statistics tables in, and then click Enter. You are then prompted to provide a password for the login name.

3. Type a password for the login name provided, and then click Enter. You are then prompted to provide a metadata prefix.

4. You can enter characters to use as a prefix for the names of your metadata tables or use no prefix, as described below:
   - Type the required prefix characters, and then click Enter.
   - Leave the prompt blank, and then click Enter to provide no metadata prefix.

   You are then prompted to provide a temp table prefix.

5. You can enter characters to use as a prefix for the names of temp tables or use no prefix, as described below:
   - Type the required prefix characters, and then click Enter.
   - Leave the prompt blank, and then click Enter to provide no temp table prefix.

   You are then prompted to provide a MicroStrategy user login.

6. Type a valid MicroStrategy user login that has administrator privileges, and then click Enter. You are then prompted to provide a password for the login name.
The default administrator account is Administrator with a blank password. This should be changed after you initial configuration.

7. Type a password for the MicroStrategy user login provided, and then click Enter. You are then prompted to choose the type of Intelligence Server configuration to complete.

8. You can perform one of the Intelligence Server configuration tasks, which are described in the sections below:

- Type 1, and then click Enter to create a new server definition. Refer to Creating and using a server definition, page 717 for steps to create a new server definition.

- Type 2, and then click Enter to use an exiting server definition. This configuration follows the same steps for creating a new server definition, which are described in Creating and using a server definition, page 717.

- Type 3, and then click Enter to delete a server definition. Refer to Deleting a server definition, page 720 for steps to delete a server definition.

- Type 4, and then click Enter to create a new server definition and use it as the default server definition. This configuration follows the same steps for creating a new server definition, which are described in Creating and using a server definition, page 717.

Creating and using a server definition

You perform the same steps to complete the following actions:
- Create a new server definition
- Create a new server definition and use it as the default server definition
- Use an existing server definition.

The action taken depends on what action you selected to complete in the procedure *To set up MicroStrategy Intelligence Server, page 716.*

This procedure assumes you have already opened the Configuration Wizard in command line mode and entered configuration information for your Intelligence Server, as described in *Setting up MicroStrategy Intelligence Server, page 715.*

---

**To create and use a server definition**

1. In the prompt that asks for a server definition name, type the name that distinguishes the server definition, and click **Enter**. You can click **Enter** without entering any information to use the default server definition. You are then prompted to choose the projects to load for the server definition.

2. Type the names of projects to load when the server definition starts, and then click **Enter**. Separate the project names with the \ character. You are then prompted to choose projects to not load for the server definition.

3. Type the names of projects to not load when the server definition starts, and then click **Enter**. Separate the project names with the \ character. You are then prompted to provide a TCP port to use for Intelligence Server.

4. You can use the default port number or enter a different port number for Intelligence Server, as described below:
- Leave the prompt blank, and then click **Enter** to use the default port number.

- Type a port number, and then click **Enter**.

  You are then prompted whether to register Intelligence Server as a service.

5. You can choose whether to register Intelligence Server as a service, as described below:

- Type **Y**, and then click **Enter** to register Intelligence Server as a service. To perform this configuration, you must be logged into your Linux machine with an account that has root level access and permissions.

- Type **N**, and then click **Enter** to not register Intelligence Server as a service.

  You are then prompted whether to start Intelligence Server when finished.

6. Type **Y** and click **Enter** to start Intelligence Server after the response file is executed. You are then prompted to provide a name for the response.ini file.

To generate a response.ini file:

1. By default, the configuration is saved as **Response.ini** in the `/HOME_PATH/` directory, where **HOME_PATH** is the directory you specified as the Home Directory during installation. You can leave the field blank to use the default name or type a different name, and then click **Enter**. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

2. You can choose to run the configuration you just completed or to run the configuration using the response.ini file at a different
time, as described below:

- **Type y**, and then click **Enter** to run the configuration.

  You can also use the **response.ini** file created for future configurations, as described in *Using the response.ini file to Configure MicroStrategy, page 727*.

- **Type n**, and then click **Enter** to quit without running the configuration. You can use the **response.ini** file created for future configurations, as described in *Using the response.ini file to Configure MicroStrategy, page 727*.

**Deleting a server definition**

You can delete a server definition to remove it from the available server definitions for Intelligence Server.

This procedure assumes you have already opened the Configuration Wizard in command line mode and entered configuration information for your Intelligence Server, as described in *Setting up MicroStrategy Intelligence Server, page 715*.

---

**To delete a server definition**

In the prompt that asks for server definitions to be removed, type the name that distinguishes the server definition, and click **Enter**. You can list multiple server definitions to be deleted, separating server definition names with the \ character. You are then prompted to provide a name for the **response.ini** file.

1. Type the name that distinguishes the server definition in the prompt that asks for server definitions to be removed. Click **Enter**.
You can list multiple server definitions to be deleted, separating server definition names with the \ character.

2. Name the response.ini file.

To generate a response.ini file

1. By default, the configuration is saved as Response.ini in the /HOME_PATH/ directory, where HOME_PATH is the directory you specified as the Home Directory during installation. You can leave the field blank to use the default name or type a different name, and then click Enter. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

2. You can choose to run the configuration you just completed or to run the configuration using the response.ini file at a different time, as described below:

   • Type Y, and then click Enter to run the configuration.

     You can also use the response.ini file created for future configurations, as described in Using the response.ini file to Configure MicroStrategy, page 727.

   • Type N, and then click Enter to quit without running the configuration. You can use the response.ini file created for future configurations, as described in Using the response.ini file to Configure MicroStrategy, page 727.

Creating a project source

If you selected option 3 in Creating a response.ini File, page 704, you can create project sources, as described in the procedure below.
To create a project source

1. In the prompt that asks for a project source name, type the name for the project source to be created, and then click Enter. You are then prompted to provide the Intelligence Server name.

2. Type the Intelligence Server name, and then click Enter. You can also click Enter without typing any information to accept the default Intelligence Server. You are then prompted to provide a TCP port to use for Intelligence Server.

3. You can use the default port number or enter a different port number for Intelligence Server, as described below:
   - Leave the prompt blank, and then click Enter to use the default port number.
   - Type a port number, and then click Enter.

4. Type a numerical value (in minutes) for the amount of inactivity that is allowed before a user is automatically disconnected from a project source. This enforces a connection time out for inactive users connected to a project source. Type 0 to define that users are not disconnected from project sources due to inactivity. Then click Enter.

5. You can type the corresponding number to select one of the authentication types listed in the command line. For information on each authentication type, see Authentication modes, page 402.
You are then prompted to provide a name for the response.ini file.

To generate a response.ini file

1. By default, the configuration is saved as Response.ini in the /HOME_PATH/ directory, where HOME_PATH is the directory you specified as the Home Directory during installation. You can leave the field blank to use the default name or type a different name, and then click Enter. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

2. You can choose to run the configuration you just completed or to run the configuration using the response.ini file at a different time, as described below:
   - Type Y, and then click Enter to run the configuration.
     
     You can also use the response.ini file created for future configurations, as described in Using the response.ini file to Configure MicroStrategy, page 727.
   - Type N, and then click Enter to quit without running the configuration. You can use the response.ini file created for future configurations, as described in Using the response.ini file to Configure MicroStrategy, page 727.

Creating a Platform Analytics project

If you selected option 1 in Creating a response file, you can choose to Create a Platform Analytics Project or Upgrade existing environment to MicroStrategy Analytics Enterprise.

You can create a Platform Analytics project as described in the procedure below.
If you choose **Create a Platform Analytics Project** when a project already exists, a message appears indicating the project exists and asks if you’d like to upgrade your Platform Analytics project. Enter `y` to continue or enter `n` to exit. If you choose **Upgrade existing environment to MicroStrategy Analytics Enterprise** when a project does not exist, a message appears indicating the project does not exists and asks if you want to create a new Platform Analytics project. Enter `y` to continue or `n` to exit.

This procedure assumes you have already opened the Configuration Wizard in command line mode and selected to create a Platform Analytics project, as described in *Creating a response.ini File, page 704*.

To create a Platform Analytics project

1. Enter your MicroStrategy username and password to connect to the Intelligence Server. You must have administrative privileges.

2. If the Configuration Wizard detects an existing Platform Analytics Project in the metadata, a message is displayed on whether to update the Platform Analytics project tables. If you type `Y` and press `Enter`, all information in the existing Platform Analytics project tables is overwritten when the response file is executed at a later time.

3. Select the location of the Project Analytics project packages.

4. Select the location of the Project Analytics configuration packages.

5. Type the number corresponding to the Mysql ODBC DSN for the database to store your Platform Analytics Project tables and click `Enter`.

If you do not have a DSN defined on your Linux machine, see *Creating a DSN for a Data Source*. 
6. Enter the login and password to your DSN and click **Enter**.

7. Enter characters to use as a prefix for the names of your metadata tables or use no prefix and click **Enter**.

To generate a response.ini file

1. By default, the configuration is saved as `Response.ini` in the `/HOME_PATH/` directory, where `HOME_PATH` is the directory you specified as the Home Directory during installation. You can leave the field blank to use the default name or type a different name, and then click **Enter**. The `response.ini` file is generated, and you are prompted whether to run the configuration immediately.

2. You can choose to run the configuration you just completed or to run the configuration using the `response.ini` file at a different time, as described below:

   - Type **Y**, and then click **Enter** to run the configuration.

     You can also use the `response.ini` file created for future configurations, as described in *Using the response.ini file to Configure MicroStrategy, page 727.*

   - Type **N**, and then click **Enter** to quit without running the configuration. You can use the `response.ini` file created for future configurations, as described in *Using the response.ini file to Configure MicroStrategy, page 727.*

To update a Platform Analytics project

1. Choose **Upgrade Platform Analytics Project**.

2. Enter your MicroStrategy username and password to connect to the Intelligence Server. You must have administrative privileges.
3. If the Configuration Wizard detects an existing Platform Analytics Project in the metadata, a message is displayed on whether to update the Platform Analytics project tables. If you type Y and press Enter, all information in the existing Platform Analytics project tables is overwritten when the response file is executed at a later time.

4. Select the location of the Project Analytics project packages.

5. Select the location of the Project Analytics configuration packages.

6. Type the number corresponding to the Mysql ODBC DSN for the database to store your Platform Analytics Project tables and click Enter.

   If you do not have a DSN defined on your Linux machine, see Creating a DSN for a Data Source.

7. Enter the login and password to your DSN and click Enter.

8. Enter characters to use as a prefix for the names of your metadata tables or use no prefix and click Enter.

To generate a response.ini file

1. By default, the configuration is saved as Response.ini in the /HOME_PATH/ directory, where HOME_PATH is the directory you specified as the Home Directory during installation. You can leave the field blank to use the default name or type a different name, and then click Enter. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

2. You can choose to run the configuration you just completed or to run the configuration using the response.ini file at a different time, as described below:
In allation and Configuration Guide

- Type Y, and then click Enter to run the configuration.

  You can also use the response.ini file created for future configurations, as described in Using the response.ini file to Configure MicroStrategy, page 727.

- Type N, and then click Enter to quit without running the configuration. You can use the response.ini file created for future configurations, as described in Using the response.ini file to Configure MicroStrategy, page 727.

Using the response.ini file to Configure MicroStrategy

This section describes how to use the response.ini file through the MicroStrategy Configuration Wizard. For information on how to configure through the MicroStrategy Configuration Wizard, see Creating a response.ini File, page 704.

1. In a Linux console window, browse to HOME_PATH, where HOME_PATH is the specified home directory during installation.

2. Browse to the folder bin.

3. Type mstrcfgwiz-editor -r ResponseFile, where ResponseFile is the full qualified path to the response.ini file. For example:

   mstrcfgwiz-editor -r /home/username/MicroStrategy/RESPONSE.INI

4. Click Enter.

If the path and the response file are valid, the configuration is performed and a successful configuration message appears. If an error occurs before or during the process, an error message displays the error that occurred when executing the response file.

Parameters and Options in the response.ini File

For a list of all parameters and options available for a response.ini file, see Response configuration parameters and options, page 408.
Configuring and Controlling Intelligence Server

MicroStrategy provides various command line tools to configure and control Intelligence Servers running on Linux. Each command line tool provides descriptive prompts and help information to guide you on how you can use the tool. This section gives a general overview of each tools functionality, and how to access more detailed information on how to use the tools.

On a Windows machine, these configurations can be completed with MicroStrategy Developer, Service Manager, and other MicroStrategy tools. For information on performing various administrative tasks, see the System Administration Guide.

Starting, Configuring, and Monitoring Intelligence Server with mstrsvr

If your Intelligence Server is installed on a Linux machine, you can start, configure, and monitor your Intelligence Server from the command line with mstrsvr. This tool starts Intelligence Server from the command line and displays the following information about your Intelligence Server connection:

- Intelligence Server version number
- Intelligence Server instance name
- Metadata DSN
- Metadata login
- Intelligence Server definition name
- Port number

You can then perform various configuration and monitoring tasks for your running Intelligence Server, which includes but is not limited to:
Display database connection information
Open, idle, and resume projects
Check and close jobs
Monitor users
Define server clustering options
Monitor memory usage information
Stop the server
Monitor lock contentions

To start, configure, and monitor Intelligence Server with mstrsvr

From a Linux console window, browse to \texttt{HOME\_PATH}, where \texttt{HOME\_PATH} is the directory that you specified as the home directory during installation.

Browse to the folder \texttt{bin}.

Type \texttt{mstrsvr}, and then click \texttt{Enter}.

Once Intelligence Server is started, general configuration information is displayed along with all available configuration and monitoring options. Perform any configuration and monitoring tasks you require.

To quit the tool and stop Intelligence Server, type \texttt{S}, and then click \texttt{Enter}.

Configuring the Default Server Instance with mstrsvr-configure

You can configure the default server instance for Intelligence Server using \texttt{mstrsvr-configure}, which is a wizard-style command line tool that prompts you for the required information.
To configure the default server instance with mstrsvr-configure

1. From a Linux console window, browse to $HOME_PATH$, where $HOME_PATH$ is the directory that you specified as the home directory during installation.

2. Browse to the folder bin.

3. Type mstrsvr-configure, and then click Enter.

4. Type a port number, and then click Enter.

5. Type a DSN, and then click Enter.

6. Type a valid login for the DSN, and then click Enter.

7. Type a valid password for the DSN login, and then click Enter.

8. Type a server definition name, and then click Enter.

Creating and Configuring Intelligence Server Instances with mstrctl

You can create and configure Intelligence Server instances with the mstrctl tool. Intelligence Servers running with a particular server definition are referred to as server instances.

To create and configure Intelligence Server instances with mstrctl

1. From a Linux console window, browse to $HOME_PATH$, where $HOME_PATH$ is the directory that you specified as the home directory during installation.

2. Browse to the folder bin.

3. Type mstrctl -h, and then click Enter.
4. Review the help information and run any required configuration tasks.

There are some commands that can output information to a file, or require a long definition that can be retrieved from a file. For information on using files to store output from and provide input to `mstrctl` commands, see *Using files to store output and provide input, page 731*.

You do not need to enter any command to quit the `mstrctl` tool because it is a one-line command line tool.

**Using files to store output and provide input**

With the `mstrctl` command line tool, you can perform the following tasks:

- Display and modify a server configuration
- Display and modify a service configuration
- Display and modify a server instance configuration

The commands that display the configurations listed above output long XML definitions to the command line. The commands that modify the configurations listed above require a long XML definition as input.

Rather than displaying and inputting long XML definitions from the command line, you can use files to store and provide input for long XML definitions.

- 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.
• Prior to using commands to display and modify service configurations (gsvc and ssvc) you must register Intelligence Server as a service. You can perform this task by using the rs command for mstrctl. To register an Intelligence Server as a service on a Linux machine, you must be logged in with an account that has root user privileges and permissions.

The following commands can have their output sent to a file:

- gsc: Displays a server configuration
- gsvc: Displays a service configuration
- gsic: Displays a server instance configuration

For example, you can run the following command to output the server instance configuration to an XML file:

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

A ServerInstance.xml file is saved in the current directory.

The following commands can read input from a file:

- ssc: Modifies a server configuration
- ssvc: Modifies a service configuration
- ssic: Modifies a server instance configuration

For example, you can run the following command to modify the server instance configuration by reading input from an XML file:

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

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

It would be difficult and time consuming to type a complete server, service, or server instance configuration from the command line. An easier way to provide this 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.

Supporting Reserved Words and Characters

When you perform MicroStrategy configuration tasks through the Linux operating system console, you must make sure reserved words and characters are not mistakenly included in your commands.

Linux operating system consoles use reserved words and characters to perform various actions. For example, the $ character may perform an action when included as part of a command executed through the operating system console. If this character is included in a command to configure, it can cause the command to fail.

For example, you use the following command to create a DSN to an Oracle database:

```
mstrconnectwiz ORCLW $MyOracleDSN 12.34.56.78 orcl 1521 -u:OracleUser -p:OracleUserPasword
```

Notice that the name of the DSN begins with the $ character. If this is a reserved character, the command fails to execute properly.

To avoid this problem, you can place single quotes (') around any character strings that may include reserved words or characters. This prevents the operating system console from interpreting the characters as an operating system action, and instead includes them as part of the character string. For example, the same command as above to create a DSN can be rewritten as follows:

```
mstrconnectwiz ORCLW 'MyOracleDSN' 12.34.56.78 orcl 1521 -u:OracleUser -p:OracleUserPasword
```

This time, the name of the DSN $MyOracleDSN is enclosed by single quotes, which allows all of the characters to be interpreted as a string of characters.
Configuring your MicroStrategy Installation

To help guide the rest of your installation and configuration steps, refer to the section *Installing and Configuring MicroStrategy on Linux, page 112* in *Chapter 1, Planning Your Installation*, for an installation and configuration checklist.
ADDING OR REMOVING MICROSTRATEGY COMPONENTS
This section describes how to add or remove MicroStrategy components on different operating systems and includes the following:

Adding or Removing MicroStrategy Components on Windows ........................................ 736
Re-installing MicroStrategy components on Windows ........................................... 737
Uninstalling MicroStrategy Components on Windows ........................................... 740
Uninstalling MicroStrategy Components on Linux ................................................. 742

Adding or Removing MicroStrategy Components on Windows

If you installed the MicroStrategy components using a disk, you will need your original installation disk to add or remove MicroStrategy components.

To add or remove MicroStrategy components

2. Open the Microsoft Control Panel and navigate to the options to add or remove programs. See the Microsoft documentation for steps to access these options.
4. Select Modify and click Next.
5. Select to accept the license agreement and click Next.
6. Verify your customer information and click Next.
7. Select the components to add by clicking their check boxes. Clear the check boxes for the components you want to uninstall and click **Next**.

   The components that are currently installed are displayed with their check boxes selected. These components are not re-installed during the modification process. If you clear any of the check boxes, that particular component is uninstalled during the modification process. You are advised not to clear the check boxes of the components that are already installed, unless you want to remove the component.

8. If prompted to stop your Web server, click **Yes**.

9. Verify the settings and click **Next**.

10. After the modification routine is complete, click **Finish**. To fully remove MicroStrategy Office, see *Uninstalling MicroStrategy Office, page 741*.

For more details on each page of the MicroStrategy Installation Wizard, see Chapter 2, *Installing MicroStrategy on Windows*.

**Re-installing MicroStrategy components on Windows**

You can re-install MicroStrategy components if there are problems with previously installed components. During re-installation the list of components previously installed are displayed and these components are re-installed. If you installed the MicroStrategy components using a disk, you need your original installation disk to repair the installation.

The re-installation of MicroStrategy Office must be performed separately. The procedure for re-installing MicroStrategy Office is explained in the following section.
To re-install MicroStrategy components


2. Open the Microsoft Control Panel and navigate to the options to add or remove programs. Refer to the appropriate Microsoft documentation for steps to access these options.


4. Select Repair and click Next.

5. Accept the license agreement and click Next.

6. You are prompted to select Yes to continue with the re-installation procedure and overwrite the components. If you do not want to overwrite the components, select No.

7. If prompted to stop your Web server, click Yes to stop it and continue with the re-installation.

8. After the re-installation routine is complete, click Finish.

For details on each page of the MicroStrategy Installation Wizard, see Chapter 2, Installing MicroStrategy on Windows.

Re-installing MicroStrategy Office

This information applies to MicroStrategy Office, the add-in for Microsoft Office applications which is no longer actively developed.

⚠️ It was substituted with a new add-in, MicroStrategy for Office, which supports Office 365 applications. The initial version does not yet have all the functionalities of the previous add-in.
For more information, see the MicroStrategy for Office page in the 2019 Update 1 Readme and the MicroStrategy for Office Online Help.

This section describes the re-installation procedure for MicroStrategy Office on Windows 7, 8.1, and 10.

To re-install MicroStrategy Office


2. Navigate to the add/remove menus for Windows:
   - For Windows 7 or 8.1 users, open the Microsoft Control Panel or navigate to the options to add or remove programs. See the appropriate Microsoft documentation for steps to access these options.
   - For Microsoft Windows 10 users, select Settings under the Start Menu and navigate to the Apps.

3. In MicroStrategy Office, follow the below instructions for your Windows version to begin re-installing:

4. Select Repair and click Next.

5. Accept the license agreement and click Next.
6. You are prompted to select **Yes** to continue with the re-installation procedure and overwrite the components. If you do not want to overwrite components, select **No**.

7. If prompted to stop your Web server, click **Yes** to stop it and continue with the re-installation.

8. After the re-installation routine is complete, click **Finish**.

For details on each page of the MicroStrategy Installation Wizard, see *Chapter 2, Installing MicroStrategy on Windows*.

**Uninstalling MicroStrategy Components on Windows**

The uninstallation procedure performs the following functions:

- Unregisters and removes selected files, registry entries, and shortcuts logged in the `Uninstall.log` file.

- Calls a custom DLL to handle unlogged items, such as registry entries and files.

Before uninstallation begins, the DLL file performs the following functions:

- Checks for user privileges. If they are not valid, uninstallation stops.

- Checks for running components. If a component is found running, uninstallation stops.

- Stops and deletes the MicroStrategy Intelligence Server service. This occurs only when the Intelligence Server is being uninstalled.

- Deletes files created by the application, such as `*.log`, `*.gid`, `*.ldb` and `*.tb`. 
The uninstallation of MicroStrategy Office must be performed separately. The procedure for uninstalling MicroStrategy Office is explained in the following sections.


2. Open the Microsoft Control Panel and navigate to the options to add or remove programs. See the appropriate Microsoft documentation for steps to access these options.


4. Select Remove and click Next.

5. Click Yes to any prompts that appear.

6. If prompted to stop your Web server, click Yes.

7. After the uninstall is complete, select Yes to restart your computer, or No to restart it later.

8. Click Finish.

You should restart the computer for a clean uninstall.

Uninstalling MicroStrategy Office

This information applies to MicroStrategy Office, the add-in for Microsoft Office applications which is no longer actively developed.

It was substituted with a new add-in, MicroStrategy for Office, which supports Office 365 applications. The initial version does not yet have all the functionalities of the previous add-in.

For more information, see the MicroStrategy for Office page in the 2019 Update 1 Readme and the MicroStrategy for Office Online Help.
This section describes the steps to uninstall MicroStrategy Office for Windows 7, 8.1, and 10.


2. Navigate to the add/remove menus for Windows:
   - For Windows 7 or 8.1 users, open the Microsoft Control Panel or navigate to the options to add or remove programs. See the appropriate Microsoft documentation for steps to access these options.
   - For Microsoft Windows 10 users, select Settings under the Start Menu and navigate to the Apps.

3. In MicroStrategy Office, follow the below instructions for your Windows version to begin uninstalling:
   - For Windows 7 or 8.1 users, select MicroStrategy Office and click Change.
   - For Windows 10 users, select MicroStrategy Office and click Uninstall.

4. Select Remove and click Next.

5. When prompted, select Yes.

6. Click Finish.

Uninstalling MicroStrategy Components on Linux

This section discusses how to uninstall MicroStrategy Intelligence Server and other MicroStrategy products on a Linux platform.

This does not apply to Platform Analytics. The Platform Analytics project must be manually deleted using any of the supported delete methods.
To uninstall MicroStrategy products on Linux

1. In a Linux console window, browse to INSTALL_PATH, where INSTALL_PATH is the directory you specified as the install directory during installation.

2. Browse to the /_uninst folder.

3. You can uninstall MicroStrategy products using the MicroStrategy Installation Wizard or through the command line as described in the options below:
   - To uninstall using the MicroStrategy Installation Wizard, type ./setup.sh and click Enter.
   - To uninstall through the command line, type ./setup.sh -console and click Enter. Follow the command line prompts to complete the uninstallation process.

4. A dialog box prompts for a language. Specify the language to be used for the MicroStrategy uninstallation and click OK to proceed to the next step.


6. Select Use an existing installation, and then select the installation from the drop-down list.

7. From the Selected Operation area, select one of the following uninstallation options:
   - **Modify**: Select this option to uninstall only some of the MicroStrategy products installed on the machine. Click Next.
   - **Uninstall**: Select this option to uninstall all MicroStrategy products installed on the machine. Click Next.
8. The License Agreement page opens. Review the license agreement and select I accept the terms of the license agreement, and then click Next. The Customer Information page opens.

9. Type your user name, company name, and license key for your installation and click Next. The Select Components page opens.

10. Clear the check box for all MicroStrategy products that are to be uninstalled, and then click Next.

You can also install products as part of the uninstallation by selecting the check box for any MicroStrategy product not previously installed.

11. On the Product Uninstallation page, select the products to uninstall. Click Next.

12. On the Start installer operation page, verify the information and click Start.

13. After uninstallation is complete, a message is displayed. Click Finish.
EXPORT ENGINE CONFIGURATION
The following sections include steps for further configuration of the Export Engine:

**Installation of Export Engine**

The Export Engine service will be installed automatically by the MicroStrategy installer when selecting to install the Intelligence server. The service will use port 20100 by default.

**Windows Installations**

For Windows installations, the Export Engine service will begin automatically after the Intelligence Server is started or after installation; however, if the server is restarted, the service must be started manually. Find and control this service through Windows Task Manager or Windows services panel, listed as MSTR_PDFExporter.

**Linux Installations**

For Linux installations, the Export Engine Service will begin automatically after the Intelligence Server is started or after installation; however, if the server is restarted, the service must be started manually. The script to start the PDF Exporter service can be found under the path `install/IntelligenceServer/PDFExportService`. The commands for executing this script are:

- `pdfexporter.sh start` to start the PDF Exporter Service.
- `pdfexporter.sh stop` to stop the PDF Exporter Service.
- `pdfexporter.sh restart` to restart the PDF Exporter Service.
- `pdfexporter.sh status` to check the status the PDF Exporter Service.
Connecting the Intelligence Server to an Export Engine on a Different Machine

You can configure the Intelligence Server to use an export engine that is installed on a different machine. You may want to do this if you are using a dedicated machine for the export engine or if you are using RHEL 6.x for your Intelligence Server, which does not support the installation of the export engine. The export engine is supported by Windows, RHEL 7.x, and SUSE 12 and 15.

To connect the Intelligence Server to an export engine on a different machine, update the registry on the Intelligence Server machine. No changes need to be made on the machine that has the export engine installed.

1. Open the registry.

On Windows, open up the Registry Editor.

On Linux, navigate to the `<HOME_PATH>` and open the file MSIReg.reg.

2. Navigate to the following location and update the registry key "host" to the Fully Qualified Name or IP Address of the machine where the export engine is located:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\MicroStrategy\DSS Server\PDFExporter]
"host"="localhost"
```

3. In the same location, update the registry key "port" to the port number that is running the export engine service.

During installation, the default value for the service is 20100. To calculate the port number, use "dword:" and the hexadecimal value of the decimal to be used.
For example, 00004E84 converted from hexadecimal to decimal is 20100, which gives you the full value of "dword:00004E84" as shown in the example below:

"[HKEY_LOCAL_MACHINE\SOFTWARE\MicroStrategy\DSS Server\PDFExporter]":
"port"=dword:00004E84

4. Restart the Intelligence Server.

Export Engine: Puppeteer Integration

The Export Engine for MicroStrategy 2019 uses the headless browser Puppeteer to export dossiers to PDF. All other formatting requests for reports and Report Services Documents are performed by the Intelligence Server.

Puppeteer requires one of the following operating systems:

Windows:
- Server 2016
- Server 2012 R2
- Server 2012
- Server 2008 R2 SP1

Linux:
- RHEL 7 or above
- SUSE 12 or above
- Amazon Linux (based on RHEL 7)
Dependencies for Linux Operating Systems

When installing MicroStrategy, the installer will run a check to ensure that all necessary packages are installed on your machine. You will be notified by the installer of any missing requirements that are needed.

Chrome Dependencies

The Puppeteer module contains Chrome binary which has certain dependencies depending on your operating system. Ensure that the following packages are installed to ensure Chrome binary functions properly.

<table>
<thead>
<tr>
<th>Chrome Dependency</th>
<th>RHEL 7 package</th>
<th>SUSE 12 Package</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>libpthread.so.0</td>
<td></td>
<td>glibc</td>
<td></td>
</tr>
<tr>
<td>libdl.so.2</td>
<td></td>
<td>glibc</td>
<td></td>
</tr>
<tr>
<td>librt.so.1</td>
<td>glibc</td>
<td>glibc</td>
<td></td>
</tr>
<tr>
<td>libm.so.6</td>
<td>libX11</td>
<td>libX11-6</td>
<td></td>
</tr>
<tr>
<td>libc.so.6</td>
<td>libX11</td>
<td>libX11-xcb1</td>
<td></td>
</tr>
<tr>
<td>ld-linux-x86-64.so.2</td>
<td>libX11</td>
<td>libX11-xcb1</td>
<td></td>
</tr>
<tr>
<td>libX11.so.6</td>
<td>libX11</td>
<td>libX11-xcb1</td>
<td></td>
</tr>
<tr>
<td>libX11-xcb.so.1</td>
<td>libX11</td>
<td>libX11-xcb1</td>
<td></td>
</tr>
<tr>
<td>libxcb.so.1</td>
<td>libxcb</td>
<td>libxcb1</td>
<td></td>
</tr>
<tr>
<td>libXcomposite.so.1</td>
<td>libXcomposite</td>
<td>libXcomposite1</td>
<td></td>
</tr>
<tr>
<td>libXcursor.so.1</td>
<td>libXcursor</td>
<td>libXcursor1</td>
<td></td>
</tr>
<tr>
<td>libXdamage.so.1</td>
<td>libXdamage</td>
<td>libXdamage1</td>
<td></td>
</tr>
<tr>
<td>libXext.so.6</td>
<td>libXext</td>
<td>libXext6</td>
<td></td>
</tr>
<tr>
<td>Chrome Dependency</td>
<td>RHEL 7 Package</td>
<td>SUSE 12 Package</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------</td>
</tr>
<tr>
<td>libXfixes.so.3</td>
<td>libXfixes</td>
<td>libXfixes3</td>
<td></td>
</tr>
<tr>
<td>libXi.so.6</td>
<td>libXi</td>
<td>libXi6</td>
<td></td>
</tr>
<tr>
<td>libXrender.so.1</td>
<td>libXrender</td>
<td>libXrender1</td>
<td></td>
</tr>
<tr>
<td>libXtst.so.6</td>
<td>libXtst</td>
<td>libXtst6</td>
<td></td>
</tr>
<tr>
<td>libgobject-2.0.so.0</td>
<td>glib2</td>
<td>libgobject-2_0-0</td>
<td></td>
</tr>
<tr>
<td>libglib-2.0.so.0</td>
<td>glib2</td>
<td>libglib-2_0-0</td>
<td></td>
</tr>
<tr>
<td>libnss3.so</td>
<td>nss</td>
<td>mozilla-nss</td>
<td></td>
</tr>
<tr>
<td>libnssutil3.so</td>
<td>nss-util</td>
<td>mozilla-nss</td>
<td></td>
</tr>
<tr>
<td>libsmime3.so</td>
<td>nss</td>
<td>mozilla-nss</td>
<td></td>
</tr>
<tr>
<td>libnspr4.so</td>
<td>nspr</td>
<td>mozilla-nspr</td>
<td></td>
</tr>
<tr>
<td>libcups.so.2</td>
<td>cups-libs</td>
<td>cups-libs</td>
<td></td>
</tr>
<tr>
<td>libdbus-1.so.3</td>
<td>dbus-libs</td>
<td>libdbus-1-3</td>
<td></td>
</tr>
<tr>
<td>libexpat.so.1</td>
<td>expat</td>
<td>libexpat1</td>
<td></td>
</tr>
<tr>
<td>libXss.so.1</td>
<td>libXScrnSaver</td>
<td>libXss1</td>
<td></td>
</tr>
<tr>
<td>libXrandr.so.2</td>
<td>libXrandr</td>
<td>libXrandr2</td>
<td></td>
</tr>
<tr>
<td>libgio-2.0.so.0</td>
<td>glib2</td>
<td>libgio-2_0-0</td>
<td></td>
</tr>
<tr>
<td>libasound.so.2</td>
<td>alsa-lib</td>
<td>libasound2</td>
<td></td>
</tr>
<tr>
<td>libpangocairo-1.0.so.0</td>
<td>pango</td>
<td>libpango-1_0-0</td>
<td></td>
</tr>
<tr>
<td>libpango-1.0.so.0</td>
<td>pango</td>
<td>libpango-1_0-0</td>
<td></td>
</tr>
<tr>
<td>libcairo.so.2</td>
<td>cairo</td>
<td>libcairo2</td>
<td></td>
</tr>
<tr>
<td>libatk-bridge-2.0.so.0</td>
<td>at-spi2-atk</td>
<td>libatk-bridge-2_0-0</td>
<td></td>
</tr>
<tr>
<td>libgtk-3.so.0</td>
<td>gtk3</td>
<td>libgtk-3-0</td>
<td></td>
</tr>
</tbody>
</table>
### Chrome Dependency | RHEL 7 Package | SUSE 12 Package | Notes
--- | --- | --- | ---
3.so.0 | | | |
libgdk_pixbuf-2.0.so.0 | gdk-pixbuf2 | libgdk_pixbuf-2-0-0 | |
libgcc_s.so.1 | libgcc | libgcc_s1 | |

### Additional Packages for Amazon Linux

Amazon Linux is based on RHEL 7 and the dependency list of RHEL 7 mentioned above can be applied to Amazon Linux. However, Amazon Linux does not contain any graphical library packages (GTK3) in the Yum repository. These packages must be installed from the CentOS/Fedora Yum repository.

<table>
<thead>
<tr>
<th>Missing Package</th>
<th>External Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>atk</td>
<td>atk-2.22.0-3.el7.x86_64.rpm</td>
</tr>
<tr>
<td>at-spi2-atk</td>
<td>at-spi2-atk-2.22.0-2.el7.x86_64.rpm</td>
</tr>
<tr>
<td>at-spi2-core</td>
<td>at-spi2-core-2.22.0-1.el7.x86_64.rpm</td>
</tr>
<tr>
<td>GConf2</td>
<td>GConf2-3.2.6-7.fc20.x86_64.rpm</td>
</tr>
<tr>
<td>libXScrnSaver</td>
<td>libXScrnSaver-1.2.2-6.fc20.x86_64.rpm</td>
</tr>
<tr>
<td>libxkbcommon</td>
<td>libxkbcommon-0.3.1-1.fc20.x86_64.rpm</td>
</tr>
<tr>
<td>libwayland-client</td>
<td>libwayland-client-1.2.0-3.fc20.x86_64.rpm</td>
</tr>
<tr>
<td>libwayland-cursor</td>
<td>libwayland-cursor-1.2.0-3.fc20.x86_64.rpm</td>
</tr>
<tr>
<td>gtk3</td>
<td>gtk3-3.10.4-1.fc20.x86_64.rpm</td>
</tr>
<tr>
<td>gdk-pixbuf2</td>
<td>gdk-pixbuf2-2.24.0-1.fc16.x86_64.rpm</td>
</tr>
</tbody>
</table>
Export Engine Configuration Properties

The configuration of the Export Engine service is handled through the `application.properties` file. The default location is:

- **Windows**: `C:\Program Files (x86)\MicroStrategy\Intelligence Server\PDFExportService`
- **Linux**: `/opt/mstr/MicroStrategy/install/IntelligenceServer/PDFExportService`

Application Core Properties

You must restart the PDF Exporter service if these properties are changed.

Changing any of the below properties will require you to restart NEE, not just the core properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>server.host</code></td>
<td>localhost</td>
<td>Used to generate the <code>baseUrl</code> used by the renderer. Do not change this value.</td>
</tr>
<tr>
<td><code>server.port</code></td>
<td>20100</td>
<td>Port number of the service URL.</td>
</tr>
<tr>
<td><code>server.context-path</code></td>
<td>/PDFExporterService</td>
<td>Context path of the service URL.</td>
</tr>
<tr>
<td><code>jvm.heapsize</code></td>
<td>1024</td>
<td>Maximum memory usage by the service process in MB.</td>
</tr>
</tbody>
</table>
## Subprocess Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>exporter.renderer.type</code></td>
<td>Puppeteer</td>
<td></td>
</tr>
<tr>
<td><code>exporter.renderer.process.max</code></td>
<td>1/4 of the number of CPU cores. (Min 2)</td>
<td>Maximum number of renderer processes.</td>
</tr>
<tr>
<td><code>exporter.renderer.path</code></td>
<td>Default is the system path.</td>
<td></td>
</tr>
<tr>
<td><code>exporter.renderer.file.folder</code></td>
<td>System temporary file directory.</td>
<td>Location of temporary files will be saved.</td>
</tr>
<tr>
<td><code>exporter.script.path.puppeteer</code></td>
<td>The full path of puppeteer.js file. Set by install script.</td>
<td></td>
</tr>
<tr>
<td><code>nodejs.exec</code></td>
<td>The full path of node executable file. Set by install script. If not set, NEE will use the global installed node.</td>
<td></td>
</tr>
</tbody>
</table>
## Renderer Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>exporter.renderer.execution_timeout</td>
<td>300</td>
<td>Time limit in seconds before killing phantomjs execution. If you change this value you must restart the service.</td>
</tr>
<tr>
<td>exporter.renderer.rendering.timeout</td>
<td>60</td>
<td>Maximum wait time before completing rendering in seconds.</td>
</tr>
<tr>
<td>exporter.renderer.resource.timeout</td>
<td>60</td>
<td>Maximum wait time for each resource request in seconds.</td>
</tr>
<tr>
<td>exporter.resource.idletime</td>
<td>500</td>
<td>Maximum wait time between two resource requests in milliseconds.</td>
</tr>
<tr>
<td>exporter.internal.web.resource</td>
<td>false</td>
<td>Whether to retrieve the web resources required during exporting from the export engine.</td>
</tr>
<tr>
<td>exporter.web.url.forced</td>
<td>false</td>
<td>This property has a higher priority than exporter.web.url.forced, exporter.web.url, and exporter.library.url. If this property value is set to true, exporter.web.url and exporter.library.url are not be used.</td>
</tr>
<tr>
<td>exporter.web.url</td>
<td>false</td>
<td>Whether exporter.web.url should always be used instead of that from Intelligence Server.</td>
</tr>
<tr>
<td>exporter.web.url</td>
<td></td>
<td>The URL of the web server which has Map and D3 plug-ins installed. This is used to support scheduled export (DS) of Map and D3 content.</td>
</tr>
</tbody>
</table>
### Property Default Value Comments

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSP web server format:</td>
<td></td>
<td>http(s)://&lt;host or IP&gt;:&lt;port&gt;/&lt;webAppName&gt;/servlet/mstrWeb</td>
</tr>
<tr>
<td>JSP web server format:</td>
<td></td>
<td>http(s)://&lt;host or IP&gt;:&lt;port&gt;/&lt;webAppName&gt;/servlet/mstrWeb</td>
</tr>
<tr>
<td>export library.url</td>
<td></td>
<td>Library server format: http(s)://&lt;host or IP&gt;:&lt;port&gt;/MicroStrategyLibrary/Application</td>
</tr>
<tr>
<td>logging.renderer.kafka</td>
<td>true</td>
<td>Whether to save the renderer log to Kafka.</td>
</tr>
<tr>
<td>logging.kafka.server</td>
<td>127.0.0.1</td>
<td>Kafka server IP. If you change this value you must restart the service.</td>
</tr>
<tr>
<td>logging.kafka.port</td>
<td>9092</td>
<td>Kafka server port. If you change this value you must restart the service.</td>
</tr>
<tr>
<td>logging.level.kafka</td>
<td>info</td>
<td>Level of log to write to Kafka. The value of the property should be error, trace, or info. If you</td>
</tr>
</tbody>
</table>
### Property Default Value Comments

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>logging.kafka.filename</td>
<td>NewExportEngine.log</td>
<td>Kafka log file name. If you change this value you must restart the service.</td>
</tr>
<tr>
<td>logging.file.filename</td>
<td>NewExportEngine.log</td>
<td>The file name created by file logger. (Fallback logging if kafka is unavailable.)</td>
</tr>
<tr>
<td>logging.file.size</td>
<td>2 MB</td>
<td>Size of log file for file logger. (The fall back logger when kafka is not available)</td>
</tr>
</tbody>
</table>

### Miscellaneous Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>exporter.pages.limit</td>
<td>1000</td>
<td>Value to limit the number of pages generated by the Export Engine. (-1 indicates no limit).</td>
</tr>
<tr>
<td>exporter.content.limit</td>
<td>200</td>
<td>Maximum JSON length request allowed in MB. The maximum value of this property is 2048.</td>
</tr>
</tbody>
</table>
Spring Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>spring.aop.auto</td>
<td>true</td>
<td>Whether to enable Spring AOP auto configuration. Changing this value is not recommended.</td>
</tr>
<tr>
<td>spring.jmx.enabled</td>
<td>true</td>
<td>Whether to enable Spring Boot JMX. Changing this value is not recommended.</td>
</tr>
<tr>
<td>logging.file.filename</td>
<td>NewExportEngine.log</td>
<td>Enable Local Resource Loading for the Export Engine</td>
</tr>
</tbody>
</table>

Enable Local Resource Loading for the Export Engine

The export engine retrieves its resources from the Web or Library server by default. If it is not possible to pull resources from the Web or Library server, you can have the export engine redirect its web resource requests to the export engine side. It then loads resources such as plugins (custom widgets and image layouts) and map keys from the resources folder that is pre-installed in the root folder of the PDF Export Service.

1. Open the application.properties file and set exporter.internal.web.resource to false. See Export Engine Configuration Properties for more information about this setting and the default location of this file.

2. Copy all plugins installed in Web to
   `<PDFExportServiceRootFolder>/resources/plugins`

3. Copy all plugins installed in Library to
   `<PDFExportServiceRootFolder>/resources/plugins`
4. Check to see if any map keys are configured in
   <WebRootFolder/Library>/WEB-INF/xml/config/mapConfig.xml. If they are, copy
   mapConfig.xml into
   <PDFExportServiceRootFolder>/resources/WEB-INF/xml/config/mapConfig.xml.

5. If there any images in <WebFolder>/images or
   <WebFolder>/VisFramework, you must copy them into
   resources/images and resources/VisFramework in the
   root folder of the PDF Export Service.

6. Restart the PDF Export Service.

See knowledge base article 483235 to troubleshoot issues with the
export engine.

When you install new plugin widgets, you must copy them to
resources/plugins in the Web, Library, and PDF Export Service
root folders.

When you configure map keys for ESRI, Google, or Geospatial
Services, you must copy mapConfig.xml to WEB-INF/xml/config in the Web, Library, and PDF Export Service root
folders.
Connecting to Databases and Data Sources

This appendix describes the configuration parameters required to connect MicroStrategy to various databases and data sources. Data Source Names (DSN) can be created using the MicroStrategy Connectivity Wizard, and in Linux, you can also configure parameters with the odbc.ini file. This appendix discusses the following topics:

- **Creating DSNs for Specific Data Sources, page 759**: Configuration information required to create a DSN for data sources available through the Connectivity Wizard.

- **Creating Database Connections in Web, page 799**: Steps to define a new database connection directly from Web for users to import data from a data source into MicroStrategy.

- **Configuring ODBC Parameters with ODBC.ini, page 802**: Configuration information required to configure data sources using the odbc.ini file in a Linux environment.

Creating DSNs for Specific Data Sources

You can create a DSN for data sources available through the Connectivity Wizard. The following table lists the information required.
for each type of data source when you create a new DSN using the Connectivity Wizard. For information on what operating systems each ODBC driver is certified for, see *Certified ODBC Drivers for MicroStrategy Intelligence Server, page 98.*

- For ODBC-specific driver details, refer to the different ODBC driver sections below the table.

- You can create a DSN from the command line version of the Connectivity Wizard in Linux. Browse to `HOME_PATH/bin`, where `HOME_PATH` is the directory you specified as the home directory during installation. In the console window, type `./mstrconnectwiz -h`, then press Enter. This command displays command line syntax and examples for different database platforms. Create your command based on the syntax and examples displayed. Once you run your command, a DSN is created in the `odbc.ini` file.

<table>
<thead>
<tr>
<th>MicroStrategy ODBC Driver for DB2 Wire Protocol for Windows and Linux, page 763</th>
<th>Driver Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Data source name</td>
<td>• Host name</td>
</tr>
<tr>
<td>• Database name</td>
<td>• Port number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MicroStrategy ODBC Driver for DB2 Wire Protocol for IBM Db2 for i for Windows and Linux, page 764</th>
<th>Driver Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Data source name</td>
<td>• IP address</td>
</tr>
<tr>
<td>• Collection</td>
<td>• Location</td>
</tr>
<tr>
<td>• Isolation level</td>
<td>• Package owner</td>
</tr>
<tr>
<td>• TCP port (446 in most cases)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MicroStrategy ODBC</th>
<th>Driver Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Data source name</td>
<td></td>
</tr>
<tr>
<td>Driver Details</td>
<td>Driver for IBM Db2 for z/OS for Windows and Linux, page 765</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>• IP address</td>
</tr>
<tr>
<td></td>
<td>• Collection</td>
</tr>
<tr>
<td></td>
<td>• Location</td>
</tr>
<tr>
<td></td>
<td>• Package collection</td>
</tr>
<tr>
<td></td>
<td>• Package owner</td>
</tr>
<tr>
<td></td>
<td>• TCP port (446 in most cases)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Driver Details</th>
<th>MicroStrategy ODBC Driver for Microsoft SQL Server for Windows and Linux, page 766</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Data source name</td>
</tr>
<tr>
<td></td>
<td>• Host name</td>
</tr>
<tr>
<td></td>
<td>• Port number</td>
</tr>
<tr>
<td></td>
<td>• Database name</td>
</tr>
<tr>
<td></td>
<td>• Use Windows NT Authentication for login ID</td>
</tr>
<tr>
<td></td>
<td>• Enable SQL Database (Azure) support</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Driver Details</th>
<th>MicroStrategy ODBC Driver for MongoDB for Windows and Linux, page 768</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Data source name</td>
</tr>
<tr>
<td></td>
<td>• Host name</td>
</tr>
<tr>
<td></td>
<td>• Port number</td>
</tr>
<tr>
<td></td>
<td>• Database name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Driver Details</th>
<th>MicroStrategy ODBC Driver for Oracle Wire Protocol for Windows and Linux, page 769</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data source name and either:</td>
</tr>
<tr>
<td></td>
<td>Standard connection:</td>
</tr>
<tr>
<td></td>
<td>• Host name</td>
</tr>
<tr>
<td></td>
<td>• Port number (in most cases, 1521)</td>
</tr>
<tr>
<td></td>
<td>• SID (MicroStrategy default is ORCL)</td>
</tr>
<tr>
<td></td>
<td>• Service name</td>
</tr>
<tr>
<td></td>
<td>• Alternate servers</td>
</tr>
<tr>
<td></td>
<td>TNSNames connection:</td>
</tr>
<tr>
<td></td>
<td>• Server name</td>
</tr>
</tbody>
</table>
### Driver Details

| MicroStrategy ODBC Driver for SAP Sybase ASE Wire Protocol for Windows and Linux | • TNSNames file  
| MicroStrategy ODBC Driver for SequeLink, page 771 | • Data source name  
| | • Network address  
| | • Database name  
| | • Enable unicode support  
| MicroStrategy ODBC Driver for Salesforce, page 776 | The MicroStrategy ODBC Driver for SequeLink allows you to access Microsoft Access databases or Microsoft Excel files stored on a Windows machine from an Intelligence Server hosted on a Linux machine.  
| | The MicroStrategy ODBC Driver for Salesforce allows you to access resources on Salesforce.com, from an Intelligence Server hosted on a Linux machine.  

This section also provides information on how to install and configure drivers from other vendors with MicroStrategy:

- **ODBC Driver for Sybase Adaptive Server IQ for Linux**
- **ODBC Driver for Teradata for Linux, page 777**
- **ODBC Driver for Informix 8 for Linux, page 779**
- **ODBC Driver for Netezza for Linux, page 781**
- **ODBC Driver for MySQL for Linux, page 783**
- **ODBC Driver for Aster Database for Linux, page 785**
- **ODBC Driver for DataDirect Cloud for Linux, page 788**
- **ODBC Driver for Vertica for Linux, page 790**
- **ODBC Driver for SAP HANA for Windows and Linux, page 792**
- **Other Data Sources and Relational Databases for Windows, page 797**
MicroStrategy ODBC Driver for Amazon Redshift Wire Protocol for Windows and Linux

The following information is required for setting up the driver connection for the MicroStrategy ODBC Driver for Amazon Redshift Wire Protocol:

- **Data Source Name**: A name to identify the Amazon Redshift data source configuration in MicroStrategy. For example, Finance or Redshift-1 can serve to identify the connection.

- **Host Name**: The server name or IP address of the machine on which the Amazon Redshift data source resides. Contact your system administrator for the server name or IP address.

- **Port Number**: The port number for the connection. In most cases, the default port number is **5439**, but you should check with your database administrator for the correct number.

- **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.

MicroStrategy ODBC Driver for DB2 Wire Protocol for Windows and Linux

The following information is required for setting up the driver connection for MicroStrategy ODBC Driver for DB2 when running against IBM Db2:

- **Data Source Name**: A name to identify the Db2 data source configuration in MicroStrategy. For example, Finance or Db2-Serv1 can serve to identify the connection.

- **Host Name**: The name of the machine that runs the Db2 server.

- **Database Name**: The name of the database to connect to by default, which is assigned by the database administrator.
Port Number: The Db2 server listener's port number. In most cases, the default port number is **50000**, but check with your database administrator for the correct number.

**MicroStrategy ODBC Driver for DB2 Wire Protocol for IBM Db2 for i for Windows and Linux**

The following information is required for setting up the driver connection for MicroStrategy ODBC Driver for DB2 Wire Protocol for IBM Db2 for i:

- **Data Source Name**: A name to identify the Db2 data source configuration in MicroStrategy. For example, Finance or Db2-1 can serve to identify the connection.

- **IP Address**: The IP Address of the machine where the catalog tables are stored. This can be either a numeric address such as **123.456.789.98**, or a host name. If you use a host name, it must be located in the **HOSTS** file of the machine or a DNS server.

- **Collection**: The name that identifies a logical group of database objects.

- **Location**: The Db2 location name, which is defined during the local Db2 installation.

- **Isolation Level**: The method by which locks are acquired and released by the system.

- **Package Owner**: The package's AuthID if you want to specify a fixed user to create and modify the packages on the database. The AuthID must have authority to execute all the SQL in the package.

- **TCP Port**: The Db2 DRDA listener process's port number on the server host machine provided by your database administrator. The default port number is usually **446**.
MicroStrategy ODBC Driver for IBM Db2 for z/OS for Windows and Linux

The following information is required for setting up the driver connection for MicroStrategy ODBC Driver for IBM Db2 for z/OS.

- **Data Source Name**: A name to identify the Db2 z/OS data source configuration in MicroStrategy. For example, Finance or Db2z/OS-1 can serve to identify the connection.

- **IP Address**: The IP Address of the machine where the catalog tables are stored. This can be either a numeric address such as 123.456.789.98, or a host name. If you use a host name, it must be located in the HOSTS file of the machine or a DNS server.

- **Collection**: The name that identifies a logical group of database objects, which is also the current schema. On Db2 z/OS, the user ID should be used as the Collection.

- **Location**: The Db2 z/OS location name, which is defined during the local Db2 z/OS installation. To determine the Db2 location, you can run the command `DISPLAY DDF`.

- **Package Collection**: The collection or location name where bind packages are created and stored for searching purposes.

- **Package Owner (Optional)**: The package's AuthID if you want to specify a fixed user to create and modify the packages on the database. The AuthID must have authority to execute all the SQL in the package.

- **TCP Port**: The Db2 DRDA listener process's port number on the server host machine provided by your database administrator. The default port number is usually **446**.
MicroStrategy ODBC Driver for Microsoft SQL Server for Windows and Linux

The following information is required for setting up the driver connection for the MicroStrategy-branded version of the Microsoft SQL Server driver:

- **Data Source Name**: A name to identify the Microsoft SQL Server data source configuration in MicroStrategy. For example, Personnel or SQLServer-1 can serve to identify the connection.

- **Host Name**: Enter the name of a SQL Server on your network. For example, if your network supports named servers, you can specify an address such as SQLServer-1. You can also specify the IP address such as 123.45.678.998. Contact your system administrator for the server name or IP address.

  Additionally, if you use named instances to distinguish SQL Server databases, you can include the named instance along with either the server name or IP address using the format `ServerName\NamedInstance` or `IPAddress\NamedInstance`. The following are examples of providing the server name for your SQL Server database:

  - SQLServer-1\Instance1
  - 123.45.678.998\Instance1

- **Port Number**: The port number for the connection. The default port number for SQL Server is usually **1433**. Check with your database administrator for the correct number.

- **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.

- **Use Windows NT Authentication for login ID**: This option is available if you are configuring your connection on Windows. Select
this check box to use Windows NT authentication to pass a user's credentials on the Windows machine to execute against a SQL Server database.

If you use Windows NT authentication with SQL Server, you must enter the Windows NT account user name and password in the Service Manager. For information on the Service Manager, see the System Administration Guide.

Inserting date data into SQL Server 2000 tables can cause errors if the system’s Regional Settings are not set properly. Ensure that the date format is defined to be in an English format.

- **Enable Azure**: Defines whether the DSN is created to support SQL Azure. Select this check box if the DSN is used to access a SQL Azure data source.

- **Enable SSL encryption**: 
  - **Windows**: Open the DSN in the ODBC Administrator and edit the Security tab so that Encryption Method is set to 'SSL' and the Validate Server Certificate is unchecked as shown below:
EncryptionMethod=1
ValidateServerCertificate=0

MicroStrategy ODBC Driver for MongoDB for Windows and Linux

The following information is required for setting up the driver connection for the MicroStrategy ODBC Driver for MongoDB:
• **Data Source Name**: A name to identify the MongoDB data source configuration in MicroStrategy. For example, Finance or MongoDB-1 can serve to identify the connection.

• **Host Name**: The server name or IP address of the machine on which the MongoDB data source resides. Contact your system administrator for the server name or IP address.

• **Port Number**: The port number for the connection. Check with your database administrator for the correct number.

• **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.

• **Schema Definition Path**: The path where configuration files that define the relational map of native data are stored. These configuration files are created in this location when first connecting to the data source and then used for subsequent connections.

**MicroStrategy ODBC Driver for Oracle Wire Protocol for Windows and Linux**

The following information is required for setting up the driver connection for MicroStrategy ODBC driver for Oracle Wire Protocol:

• **Data Source Name**: Enter a name to identify the Oracle data source configuration in MicroStrategy. For example, Finance or Oracle-1 can serve to identify the connection. A DSN is required for any Oracle Wire Protocol connection. Depending on whether you want to use a standard connection or a TNSNames connection, refer to one of the following lists of options below:

• **Standard Connection**: A standard connection is configured through Oracle Wire Protocol with the following connection parameters:

• **Host Name**: The name of the Oracle server to be accessed. For example, the server could be named Oracle-1 or an IP address
such as 123.456.789.98.

- **Port Number**: The Oracle listener port number provided by your database administrator. The default port number is usually **1521**.

- One of the following parameters; which one you choose is up to your personal preference:

  - **SID**: The Oracle System Identifier for the instance of Oracle running on the server. The default SID is usually **ORCL**.

  - **Service Name**: The global database name, which includes the database name and the domain name. For example, if your database name is **finance** and its domain is **business.com** the service name is **finance.business.com**.

  - **Alternate Servers**: A list of alternate database servers to enable connection failover for the driver. If the primary database server entered as the SID or service name is unavailable, a connection to the servers in this list is attempted until a connection can be established. You can list the servers in SID or service name format, as shown in the following examples:

    - Using an SID: **(HostName=DB_server_name:PortNumber=1526:SID=ORCL)**

    - Using a Service Name: **(HostName=DB_server_name:PortNumber=1526:ServiceName=service.name.com)**

- **TNSNames Connection**: A TNSNames connection uses a **TNSNAMES.ORA** file to retrieve host, port number, and SID information from a server (alias or Oracle net service name) listed in the **TNSNAMES.ORA** file. A TNSNames connection requires the following parameters:

  - **Server Name**: A server name, which is included in a **TNSNAMES.ORA** file included in the TNSNames File text box below.
- **TNSNames File**: The location of your `TNSNAMES.ORA` file. Make sure to enter the entire path to the `TNSNAMES.ORA` file, including the file name itself. You can specify multiple `TNSNAMES.ORA` files.

When connecting to an installation of Oracle as metadata, it is imperative that the ODBC driver is configured to use the character set parameters as the Oracle metadata installation. This is controlled through the `IANAAppCodePage` option in the `odbc.ini` file.

Follow these steps to determine the character set of your Oracle metadata installation and adjust the `odbc.ini` files:

1. Open the MicroStrategy ODBC Test Tool and connect to your database.
2. Run the following query `select * from NLS_DATABASE_PARAMETERS`.
3. Note the value returned for the `NLS_CHARACTERSET` parameter.
5. Open the `odbc.ini` file and set the `IANAAppCodePage` value to the assigned MIBenum value from the IANA registry.

### MicroStrategy ODBC Driver for SequeLink

The MicroStrategy ODBC Driver for SequeLink allows you to access Microsoft Access databases or Microsoft Excel files stored on a Windows machine from an Intelligence Server hosted on a Linux machine. The steps below show you how to perform the necessary configurations on the various machines to support this type of configuration.
Preparing the Microsoft Access Database from an Intelligence Server Hosted on Linux

You must complete the steps below to access an Access database stored on a Windows machine from an Intelligence Server hosted on a Linux machine.

On the Windows machine where the Access database is stored, you must create a DSN to connect to the Access database. For instructions on creating a DSN, see Creating a DSN for a data source, page 363.

1. On the Windows machine that stores the Access database to connect to, install the SequeLink ODBC Socket Server. This can be installed as part of a MicroStrategy installation, and is included in the Other components options of the MicroStrategy Product Suite (see Select Features, page 122).

   The SequeLink ODBC Socket Server that is provided with a MicroStrategy installation is for exclusive use with the MicroStrategy Product Suite. You are not licensed to use this product with any application other than MicroStrategy products.

2. On the Windows machine where you installed the SequeLink ODBC Socket Server, go to the SequeLink Management Console Snap-in.

3. Under Console Root, go to SequeLink 6.0 Manager > Connected to SLAgent55 > SequeLink Services > SLSocket55 > Configuration > Data Source Settings.

4. From the Action menu, go to New > Data Source.

5. Type a descriptive name for the new data source, such as Access Data Source.

6. Expand the new data source and select Advanced.
7. Right-click **DataSourceSOCODBCConnStr** and select **Properties**. The DataSourceSOCODBCConnStr Properties dialog box opens.

8. In the **Value** field, type \texttt{DSN=AccessDSN}, where \texttt{AccessDSN} is the DSN you created to connect to your Access database. This is different from the data source you created as part of the steps to configure the SequeLink ODBC Socket Server.

9. Click **OK**.

10. Within the same data source, select **User Security**.

11. Right click **DataSourceLogonMethod** and select **Properties**.

12. From the **Value** drop-down list, select **Anonymous**.

13. Click **OK**.

14. Right-click the data source, point to **All Tasks**, and select **Save configuration**.

15. On the Linux machine that hosts your Intelligence Server, you must configure the MicroStrategy ODBC driver for SequeLink to connect to the Access database. For instructions on how to perform this configuration, see *Configuring the MicroStrategy ODBC Driver for SequeLink*, page 775.

### Preparing the Microsoft Excel File from an Intelligence Server Hosted on Linux

You must complete the steps below to access Excel files stored on a Windows machine from an Intelligence server hosted on a Linux machine.

If the Windows machine where the Excel file is stored, you must prepare the Excel file as a valid data source. For instructions to prepare an Excel
On the Windows machine where the Excel file is stored, you must create a DSN to connect to the Excel file. For instructions to create a DSN for an Excel file, see *Use your Excel file as a data source, page 798.*

1. On the Windows machine that stores the Excel files to connect to, install the SequeLink ODBC Socket Server. This can be installed as part of a MicroStrategy installation, and is included in the Other components options of the MicroStrategy Product Suite (see *Select Features, page 122*).

2. On the Windows machine where you installed the SequeLink ODBC Socket Server, go to the *SequeLink Management Console Snap-In.*

3. Under *Console Root*, expand *SequeLink 6.0 Manager*, expand *Connected to SLAgent55*, expand *SequeLink Services*, expand *SLSocket55*, expand *Configuration*, and then select *Data Source Settings.*

4. From the *Action* menu, point to *New*, and select *Data Source*. A new data source is created underneath Data Source Settings.

5. Type a descriptive name for the new data source, such as *Excel Data Source*.

6. Expand the new data source, and select *Advanced*.

7. Right-click *DataSourceSOCODBCConnStr* and select *Properties*. The DataSourceSOCODBCConnStr Properties dialog box opens.

8. In the *Value* field, type \texttt{DSN=ExcelDSN}, where *ExcelDSN* is the DSN you created to connect to your Excel file. This is different from that data source you created as part of the steps to configure the SequeLink ODBC Socket Server.

9. Click *OK.*
10. Within the same data source, select **User Security**.

11. Right click **DataSourceLogonMethod** and select **Properties**. The DataSourceLogonMethod Properties dialog box opens.

12. From the **Value** drop-down list, select **Anonymous**. This allows connection to the Excel file without using a user name and password.

13. Click **OK**.

14. Right-click the data source, point to **All Tasks**, and select **Save configuration**.

15. On the Linux machine that hosts your Intelligence Server, you must configure the MicroStrategy ODBC driver for SequeLink to connect to the Excel files. For instructions on how to perform this configuration, see *Configuring the MicroStrategy ODBC Driver for SequeLink, page 775*.

Configuring the MicroStrategy ODBC Driver for SequeLink

The steps below show you how to configure the MicroStrategy ODBC driver for SequeLink to access either Microsoft Access databases or Excel files stored on a Windows machine:

1. On the Linux machine that hosts Intelligence Server, browse to `HOME_PATH` where `HOME_PATH` is the directory you specified as the Home Directory during installation.

2. Open the `odbc.ini.example` file and find the section that starts with `[SequeLinkODBC]`. Copy this section into the `odbc.ini` file. For information on the parameters, refer to DataDirect's documentation at [http://media.datadirect.com/download/docs/odbc/allodbc/help.html](http://media.datadirect.com/download/docs/odbc/allodbc/help.html).
3. Edit the parameters listed below:

   - **Host**: Type the IP address of the Windows machine that stores the Access database or Excel files.

   - **ServerDataSource**: Type the name of the data source for the Access database or Excel file to connect to as a data source. This is the name of the data source that you defined while configuring the Sequelink ODBC Socket Server, as shown in the example image below:

     ![Diagram of Sequelink ODBC Socket Server configuration]

     - **Console Root**
       - **Sequelink 5.5 Manager (localhost)**
       - **Sequelink Services**
         - **SLagent55 [active]**
         - **SLsocket55 [active]**
       - **Configuration**
         - **Service Settings**
         - **Data Source Settings**
           - **Default**
           - **Access Data Source**
         - **Profiles**
       - **Monitor**
       - **Service Event Trace**
     - **Service Templates**

4. **Save the `odbc.ini` file.**

5. **Restart Intelligence Server.**

**MicroStrategy ODBC Driver for Salesforce**

The MicroStrategy ODBC Driver for Salesforce allows you to access resources on Salesforce.com, from an Intelligence Server hosted on a Windows or Linux machine.

You can also use MicroStrategy Web and Import Data to integrate Salesforce.com data into MicroStrategy. For steps to configure a connection to Salesforce.com to support this type of integration, see **Configuring Third-Party Data Sources for Importing Data, page 540.**
The following information is required for setting up the driver connection for the MicroStrategy ODBC driver for Salesforce:

- **Data Source Name**: A name to identify the Salesforce data source configuration in MicroStrategy. For example, Finance or Salesforce-1 can serve to identify the connection.

- **Host Name**: The URL used to log in to the Salesforce.com system. You can keep the default of login.salesforce.com to connect to the production instance. However, you can also connect to other systems such as test.salesforce.com if you are connecting to testing environments.

If you attempt to test the connection to your Salesforce.com system, the password syntax is `PasswordSecurityToken`, where `Password` is the password for the user account and `SecurityToken` is the additional security token required to access Salesforce.com. Do not use any spaces or other characters to separate the password and security token. As part of configuring a connection to your Salesforce.com system, you can include the password and security token as part of the database login, which is a component of a database instance used to access the DSN in MicroStrategy. For steps to create a database login, which you can use to provide the Salesforce.com password and security token, see *Creating a database login*.

**ODBC Driver for Teradata for Linux**

ODBC driver for Teradata is not a MicroStrategy-branded driver. The following steps show how to configure the ODBC driver for Teradata.

For information on setting up an ODBC driver for Teradata through the Connectivity Wizard on Windows, see *Other Data Sources and Relational Databases for Windows, page 797*. 
To Configure the ODBC Driver for Teradata

Install the ODBC Driver for Teradata for the correct operating system. For information on installation, refer to the product documentation provided directly by the database vendor.

The directory where the ODBC driver for Teradata is installed should always be accessible to MicroStrategy Intelligence Server.

You can enable Teradata Parallel Transporter for your connections to Teradata. This can improve performance when retrieving large amounts of data, typically 0.5 Gigabytes and larger, which can occur most commonly in MicroStrategy when publishing Intelligent Cubes. For steps to configure this support, refer to the MicroStrategy Web Help.

To Configure a DSN

1. In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation.

2. Open the odbc.ini.example file and find the section that starts with [TERADATA_SERVER]. Copy that section into the odbc.ini file in the [ODBC Data Sources] section.

3. Edit the DSN parameters DBCName, Database, and DefaultDatabase, and modify the value of MSTR_TERADATA_PATH with the location of the directory where the ODBC Driver for Teradata is installed.

You can also edit the parameters in the odbc.ini file to reflect your environment. To support parameterized queries, define the EnableExtendedStmtInfo parameter as EnableExtendedStmtInfo=Yes. For information on the other available parameters, refer to your third-party Teradata driver documentation. This can often be found along with the driver installation.
4. Save the `odbc.ini` file.

**ODBC Driver for Informix 8 for Linux**

The MicroStrategy ODBC Driver for Informix 8 is already installed in the `INSTALL_PATH/lib` directory.

- The ODBC Driver for Informix 8 for Linux is a MicroStrategy-branded ODBC driver, but it is not accessible through the Connectivity Wizard.

However, the Informix Client Software Developer's Kit (CSDK) must be installed before you create a DSN. This software is not included in the MicroStrategy product suite installation and must be obtained through the database vendor or a third party. For information on installation, refer to the product documentation provided directly by the database vendor.

The following steps show how to configure the MicroStrategy ODBC driver for Informix 8.

**To configure ODBC Driver for Informix 8**

Begin by installing the Informix CSDK.

- The directory where CSDK is installed should always be accessible to Intelligence Server.

The following set of instructions are divided into two sections: how to configure the environment and how to configure a DSN.

**To Configure the Environment**

1. In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `env`.
2. Add Write privileges to the `ODBC.sh` file by entering the following command:
   ```bash
   chmod u+w ODBC.sh
   ```

3. Edit the `ODBC.sh` file and make the following changes:
   - Provide the location of the directory where the Informix Client Software Developer's Kit (CSDK) is installed. The following definition is included:
     ```bash
     INFORMIXDIR='<INFORMIXDIR>'
     ```
     Replace this `<INFORMIXDIR>` placeholder with the directory path. Do not modify any other occurrences of `<INFORMIXDIR>` within `odbc.sh`.
   - Provide the name of the Informix Server. The following definition is included:
     ```bash
     INFORMIXSERVER='<INFORMIXSERVER>'
     ```
     Replace this `<INFORMIXSERVER>` placeholder with the directory path. This value is chosen from the list in `<INFORMIXDIR>/etc/sqlhosts`. Do not modify any other occurrences of `<INFORMIXSERVER>` within `odbc.sh`.

4. Save the `ODBC.sh` file and remove Write privileges from the file by entering the following command:
   ```bash
   chmod a-w ODBC.sh
   ```

To Configure a DSN

1. In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.
2. Open the odbc.ini.example file and search for the section that starts with [IBM INFORMIX]. Copy the section into the odbc.ini file.

3. Edit the DSN parameters Database, Servername, and Service. For information on the available parameters, refer to your third-party Teradata driver documentation. This can often be found along with the driver installation.

4. Save the odbc.ini file.

For details on these DSN parameters, refer to the product documentation provided directly by the database vendor.

**ODBC Driver for Netezza for Linux**

ODBC driver for Netezza is not a MicroStrategy-branded driver. The following steps show how to configure ODBC driver for Netezza.

You must modify odbcinst.ini file and odbc.ini file to create the DSN for Netezza.

**To Configure ODBC Driver for Netezza**

Begin by installing the ODBC Driver for Netezza for the correct operating system. For information on installation, refer to the product documentation provided directly by the database vendor.

> The directory where Netezza is installed should always be accessible to MicroStrategy Intelligence Server.

The following set of instructions are divided into two sections: how to modify the odbcinst.ini file and how to modify the odbc.ini file.
To Modify the odbcinst.ini File

1. In a Linux console window, browse to HOME\_PATH, where HOME\_PATH is the directory you specified as the home directory during installation.

2. Edit the odbcinst.ini file and replace all instances of <NETEZZA\_ODBC\_DIR> with the location of the directory where the Netezza ODBC Driver is installed. An example of this is as follows:

If the original path is:

Driver = /<NETEZZA\_ODBC\_DIR>/lib64/libnzodbc.so

Then the modified path will be:

Driver = /usr/odbc/netezzahome/lib64/libnzodbc.so

3. Save the odbcinst.ini file.

To Modify the odbc.ini File

1. Open the odbc.ini.example file and search for the section that starts with [IBM \_NETEZZA].


3. Copy and paste the contents from the odbc.ini.example file for your Netezza ODBC driver. You should paste the contents of the DSN exactly as they appear in the example file.

4. Make the following changes to the copied sample file:
   - Modify the driver location to match the location of the installed Netezza ODBC Driver.
   - Change the database, server name, user name, and password, and any other relevant parameters to match the information for your database. For information on the available parameters,
refer to your third-party Netezza driver documentation. This can often be found along with the driver installation.

5. Save the `odbc.ini` file.

For details on these DSN parameters, refer to the product documentation provided by the database vendor.

**ODBC Driver for MySQL for Linux**

The ODBC driver for MySQL is not a MicroStrategy-branded driver. The following steps show how to configure the ODBC driver for MySQL, which is certified for the Linux operating system.

You must modify the `odbc.ini` file to create the DSN for MySQL.

The third-party product(s) discussed in the procedure below is manufactured by vendors independent of MicroStrategy. MicroStrategy makes no warranty, express, implied or otherwise, regarding this product, including its performance or reliability.

**To configure ODBC driver for MySQL**

Begin by installing the 64-bit ODBC Driver for MySQL for the Linux operating system, found at the hyperlink [http://dev.mysql.com/downloads/connector/odbc/](http://dev.mysql.com/downloads/connector/odbc/). This site is valid as of the release of this manual. For information on installation, refer to the product documentation provided by the database vendor.

- Ensure that the driver files are installed to the `/usr/lib` directory.

- For exact version numbers of MySQL drivers certified with MicroStrategy, refer to the MicroStrategy General Information Readme.

The following set of instructions are divided into two sections: how to define the location of driver files and how to modify the `odbc.ini` file.
To Define the Location of the Driver Files

1. In a Linux console window, browse to HOME_PATH, where HOME_PATH is the directory you specified as the Home Directory during installation. Browse to the folder env.

2. Add Write privileges to the ODBC.sh file by entering the following command:
   chmod u+w ODBC.sh

3. Edit the ODBC.sh file and provide the location of the directory where the MySQL driver is installed. Within the ODBC.sh file, the following definition is included:
   
   MySQL_PATH='<MYSQL_PATH>'

   Replace this <MYSQL_PATH> placeholder with the location of where the MySQL driver is installed. Do not modify any other occurrences of <MYSQL_PATH> within odbc.sh.

4. Save the ODBC.sh file and remove Write privileges from the file by entering the following command:
   chmod a-w ODBC.sh

To Modify the odbc.ini File

1. In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation.

2. Open the odbc.ini.example file and search for the section that starts with [MySQL].


4. Copy and paste the contents from the odbc.ini.example file for your MySQL ODBC driver. You should paste the contents of the DSN exactly as they appear in the example file.
5. Make the following changes to the copied sample file:

- Modify the `<MYSQL_ODBC_DIR>` placeholder in the driver location to match the location of the installed MySQL ODBC Driver.
- Change the database, server name, user name, password, and any other relevant parameters to match the information for your database. For information on the available parameters, refer to your third-party MySQL driver documentation. This can often be found along with the driver installation.

  ! Warning: Ensure that there is no white space between the equals sign (=) which separates the parameter and its value.


You can test a connection to your MySQL database with the MicroStrategy DB Query Tool.

This completes the steps to create a DSN and configure an ODBC driver for MySQL. To create a database instance and database connection, see *Creating a database instance, page 435* and *Creating a database connection, page 439*.

**ODBC Driver for Aster Database for Linux**

The ODBC driver for Aster Database is not a MicroStrategy-branded driver. The following steps show how to configure the ODBC driver for Aster Database for Linux.

You must modify the `odbc.ini` file to create the DSN for Aster.

  ! Warning: The third-party product(s) discussed in the procedure below is manufactured by vendors independent of MicroStrategy. MicroStrategy makes no warranty, express, implied or otherwise, regarding this product, including its performance or reliability.
To Configure ODBC Driver for Aster Database

Begin by installing the Aster ODBC Driver for the Linux operating system. For information on installation, refer to the product documentation provided by the database vendor.

- The path to the installation location you choose for the ODBC driver is used later in this procedure as the value for the Driver parameter in the odbc.ini file.
- For exact version numbers of Aster drivers certified with MicroStrategy, refer to the MicroStrategy General Information Readme.

The following set of instructions are divided into two sections: how to configure the environment and how to modify the odbc.ini file.

To Configure the Environment

1. In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation. Browse to the folder env.

2. Add Write privileges to the ODBC.sh file by entering the following command:
   
   chmod u+w ODBC.sh

3. Edit the ODBC.sh file and provide the location of the Aster library installation. Within the ODBC.sh file, the following definition is included:

   ASTER_PATH='<ASTER_PATH>'

   Replace this <ASTER_PATH> placeholder with the location of the Aster library installation. Do not modify any other occurrences of <ASTER_PATH> within odbc.sh.

4. Save the ODBC.sh file and remove Write privileges from the file
by entering the following command:

```bash
chmod a-w ODBC.sh
```

To Modify the `odbc.ini` File

1. In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.

2. Open the `odbc.ini.example` file and search for the section that starts with `[Aster Database]`.


4. Copy and paste the contents from the `odbc.ini.example` file for your Aster ODBC driver. You should paste the contents of the DSN exactly as they appear in the example file.

5. Make the following changes to the copied sample file:
   
   - Modify the driver location to match the location of the installed Aster ODBC Driver.
   
   - Change the database, server name, user name, password, and any other relevant parameters to match the information for your database. For information on the available parameters, refer to your third-party Aster Database driver documentation. This can often be found along with the driver installation.

   ![](danger.png)

   Ensure that there is no white space between the equals sign (=) which separates the parameter and its value.


This completes the steps to create a DSN and configure an ODBC driver for Aster Database.
ODBC Driver for DataDirect Cloud for Linux

The ODBC driver for DataDirect Cloud is not a MicroStrategy-branded driver. The following steps show how to configure the ODBC driver for DataDirect Cloud for Linux.

The third-party product(s) discussed in the procedure below is manufactured by vendors independent of MicroStrategy. MicroStrategy makes no warranty, express, implied or otherwise, regarding this product, including its performance or reliability.

To Configure ODBC Driver for DataDirect Cloud

Begin by installing the DataDirect Cloud ODBC Driver for the Linux operating system. For information on installation, refer to the product documentation provided by the database vendor.

- The path to the installation location you choose for the ODBC driver is used later in this procedure as the value for the Driver parameter in the odbc.ini file.

- For exact version numbers of DataDirect Cloud drivers certified with MicroStrategy, refer to the MicroStrategy General Information Readme.

The following set of instructions are divided into two sections: how to configure the environment and how to modify the odbc.ini file.

To Configure the Environment

1. In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation. Browse to the folder env.

2. Add Write privileges to the ODBC.sh file by entering the following command:
chmod u+w ODBC.sh

3. Edit the ODBC.sh file and provide the location of the DataDirect Cloud ODBC driver files. Within the ODBC.sh file, the following definition is included:

```
DataDirectCloud_PATH='<$DataDirectCloud_PATH>'
```

Replace this `<DataDirectCloud_PATH>` placeholder with the location of the DataDirect Cloud ODBC driver files. Do not modify any other occurrences of `<DataDirectCloud_PATH>` within odbc.sh.

4. Save the ODBC.sh file and remove Write privileges from the file by entering the following command:

```
chmod a-w ODBC.sh
```

To Modify the odbc.ini File

1. In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation.

2. Open the odbc.ini.example file and search for the section that starts with [DataDirect Cloud].


4. Copy and paste the contents from the odbc.ini.example file for your DataDirect Cloud ODBC driver. You should paste the contents of the DSN exactly as they appear in the example file.

5. Make the following changes to the copied sample file:

   - Modify the driver location to match the location of the installed DataDirect Cloud ODBC Driver.
   - Change the database, server name, user name, password, and any other relevant parameters to match the information for your
database. For information on the available parameters, refer to your third-party DataDirect Cloud driver documentation. This can often be found along with the driver installation.

Ensure that there is no white space between the equals sign (=) which separates the parameter and its value.


This completes the steps to create a DSN and configure an ODBC driver for DataDirect Cloud.

**ODBC Driver for Vertica for Linux**

The ODBC driver for Vertica is not a MicroStrategy-branded driver. The following steps show how to configure the ODBC driver for Vertica for Linux.

You must modify the odbc.ini file to create the DSN for Vertica.

The third-party product(s) discussed in the procedure below is manufactured by vendors independent of MicroStrategy. MicroStrategy makes no warranty, express, implied or otherwise, regarding this product, including its performance or reliability.

**To Configure ODBC Driver for Vertica**

Begin by installing the ODBC Driver for Vertica for the Linux operating system. For information on installation, refer to the product documentation provided by the database vendor.

The path to the installation location you choose for the ODBC driver is used later in this procedure as the value for the Driver parameter in the odbc.ini file.

The following set of instructions are divided into two sections: how to configure the environment and how to modify the odbc.ini file.
To Configure the Environment

1. In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `env`.

2. Add Write privileges to the `ODBC.sh` file by entering the following command:
   ```bash
   chmod u+w ODBC.sh
   ```

3. Edit the `ODBC.sh` file and provide the location of the `vertica.ini` file. Within the `ODBC.sh` file, the following definition is included:
   ```bash
   VERTICAINI='<VERTICAINI_PATH>'
   ```

   Replace this `<VERTICAINI_PATH>` placeholder with the location of the `vertica.ini` file. Do not modify any other occurrences of `<VERTICAINI_PATH>` within `odbc.sh`.

4. Save the `ODBC.sh` file and remove Write privileges from the file by entering the following command:
   ```bash
   chmod a-w ODBC.sh
   ```

To Modify the `odbc.ini` File

1. In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.

2. Open the `odbc.ini.example` file and search for the section that starts with `[HP VERTICA]`.


4. Copy and paste the contents from the `odbc.ini.example` file for your Vertica ODBC driver. You should paste the contents of the DSN exactly as they appear in the example file.
5. Make the following changes to the copied sample file:

- Modify the driver location to match the location of the installed Vertica ODBC Driver.

- Change the database, server name, user name, password, and any other relevant parameters to match the information for your database. For information on the available parameters, refer to your third-party Vertica driver documentation. This can often be found along with the driver installation.

   ! Ensure that there is no white space between the equals sign (=) which separates the parameter and its value.


This completes the steps to create a DSN and configure an ODBC driver for Vertica.

**ODBC Driver for SAP HANA for Windows and Linux**

The ODBC driver for SAP HANA is not a MicroStrategy-branded driver. The following steps show how to configure ODBC driver for SAP HANA.

MicroStrategy recommends that the SAP HANA user account used to create the database is granted full permissions for the database. If the database user account cannot be granted full permissions to the database, you can use the recommendations listed in Required database permissions to create metadata, History List, and statistics repositories, page 376 to determine the required permissions for the SAP HANA database user account. In addition, ensure the following permissions are defined for your SAP HANA user account:

Insert permission for the `_SYS_BIC` schema.

Select permission for the `_SYS_REPO` schema.
To Configure an ODBC Driver for SAP HANA on Windows

1. Install the SAP HANA ODBC driver files on the Windows system that will host the MicroStrategy Intelligence Server. For specific installation steps, refer to your third-party SAP documentation.

2. Using the Microsoft ODBC Data Source Administrator, create a data source name to connect to your SAP HANA data source.

   For best practices on using the Microsoft ODBC Data Source Administrator to create data source names that are to be used in MicroStrategy, see *Managing ODBC and data sources with Microsoft ODBC Data Source Administrator, page 365.*

3. You can use the MicroStrategy DB Query Tool to test whether data can be retrieved data from your SAP HANA data source. For information on how to use the MicroStrategy DB Query Tool, see *Using the DB Query Tool, page 701.*

4. To use an SAP HANA as a data source, you must create a database instance in MicroStrategy. For information on creating a database instance, see *Creating a database instance, page 435.*

   When creating a database connection, which is part of a database instance, ensure that you select Non UTF-8 as the character set encoding for Windows drivers.

This completes the steps for the initial connection to SAP HANA in MicroStrategy for Windows environments. For additional configuration requirements, see *Additional Requirements to Support SAP HANA, page 796.*
To Configure an ODBC Driver for SAP HANA on Linux

Begin by installing the SAP HANA ODBC driver files on the Linux system that will host the MicroStrategy Intelligence Server. For specific installation steps, refer to your third-party SAP documentation.

The following set of instructions are divided into two sections: how to configure your environment and how to configure a DSN.

To Configure the Environment

1. In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation. Browse to the folder env.

2. Add Write privileges to the ODBC.sh file by entering the following command:
   chmod u+w ODBC.sh

3. Edit the odbc.sh file and provide the location where you installed the SAP HANA ODBC driver files. Within the ODBC.sh file, the following definition is included:
   SAPHANA_PATH='<SAPHANA_PATH>'

   Replace this <SAPHANA_PATH> placeholder with the location of the SAP HANA ODBC driver files. Do not modify any other occurrences of <SAPHANA_PATH> within odbc.sh.

4. Save the ODBC.sh file and remove Write privileges from the file by entering the following command:
   chmod a-w ODBC.sh
To Configure a DSN

1. In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation.

2. Open the odbc.ini.example file and find the section that starts with [SAP_HANA]. Copy the section into the odbc.ini file.

3. Edit the following information from the syntax that you copied to odbc.ini:
   - Driver=<SAPHANA_PATH>/libodbcHDB.so
     Replace <SAPHANA_PATH> with the location where you installed the SAP HANA ODBC driver files.
   - Servernode=ip_address:port
     Replace ip_address with the IP address for the machine that hosts the SAP HANA database.
     Replace port with the port number for the connection to the SAP HANA database. Contact your SAP HANA database administrator for the required port number.
   - USER=uid
     Replace uid with a valid SAP HANA user account.
   - PASSWORD=pwd
     Replace pwd with the password for the SAP HANA user account described above.

   For information on the available parameters, refer to your third-party SAP HANA driver documentation. This can often be found along with the driver installation.

4. Save the odbc.ini file.
5. You can use the MicroStrategy DB Query Tool to test whether data can be retrieved data from your SAP HANA data source. For information on how to use the MicroStrategy DB Query Tool, see *Using the DB Query Tool, page 701*.

6. To use an SAP HANA as a data source, you must create a database instance in MicroStrategy. For information on creating a database instance, see *Creating a database instance, page 435*.

   When creating a database connection, which is part of a database instance, for SAP HANA, ensure that you select Non UTF-8 as the character set encoding for Linux drivers.

This completes the steps for the initial connection to SAP HANA in MicroStrategy for Windows environments. For additional configuration requirements, see *Additional Requirements to Support SAP HANA, page 796* below.

**Additional Requirements to Support SAP HANA**

Review the following additional requirements to ensure a successful integration of SAP HANA in MicroStrategy:

- Be aware that once you import tables from SAP HANA into MicroStrategy, you must manually import any prefix information as well. Using the Warehouse Catalog, you can select all tables imported from SAP HANA and select Import Prefix to import the prefix information. For additional steps to access and use the Warehouse Catalog, see the *Project Design Guide*.

- If the tables in SAP HANA include input parameters, these are supported in MicroStrategy using prompts. Using the Table Editor available in MicroStrategy Developer, you can create and modify prompts to support input parameters. For steps to access and use the Table Editor, refer to the *Project Design Help*.
Other Data Sources and Relational Databases for Windows

If you use other databases or data sources, refer to the database-specific documentation for information on required settings. Standard settings are supported by MicroStrategy for most relational databases.

Microsoft Excel

A Microsoft Excel file can be used as a data source in MicroStrategy. The information below explains how to prepare an Excel file for use with MicroStrategy and how to connect to the Excel file.

This data can be used as part of a MicroStrategy project in various ways. For example, you can integrate the Excel data in your project using tools such as Architect, as described in the Project Design Guide. You can also use Freeform SQL and Query Builder to access your Excel data, as described in the Advanced Reporting Guide.

Prepare an Excel file as a valid data source

To use an Excel file as a data source, you must create and store the data in the Excel file so that it can be recognized in MicroStrategy as a set of tables that contain valid data. Follow the below instructions to create a table with valid data in an Excel file:

1. Prepare the Excel file as follows:
   
   - Ensure that all column headers are of a valid format:
     
     - No spaces in the header name (for example, Category_ID instead of Category ID).
     
     - Alphanumeric, and beginning with a letter.
     
     - Ensure that all cells for the ID column have a value in them.
2. In the Excel file, create a table by performing the following:

   a. Highlight the specific rows and columns with the data to use to create a report with, including the column headers, such as Category_ID and Category_DESC.

   Do not use the column headings at the top of the Excel spreadsheet, marked as A, B, C, and so on to select the whole column. Doing so may include numerous empty cells with NULL values.

   b. In the Name Box, type a name for the highlighted cells, and then press ENTER. The name you type in is used in MicroStrategy as a table name.

      The Name Box is the drop-down list on the left-hand side below the toolbars.

      You can create multiple tables in one Excel file by highlighting different parts of the file and assigning them different names.


   Ensure that the file is not password-protected.

Use your Excel file as a data source

To use an Excel file as a data source, you can create a data source name (DSN) for the Excel file. This DSN can be used by a database instance in MicroStrategy to connect to the Excel file. For information on creating a database instance, see Creating a database instance, page 435.

As an alternative, you can use Data Import to quickly include Excel data in your MicroStrategy project. Steps to use Data Import to import
data and begin your analysis is included in the *MicroStrategy Web Help*.

Text files

A text file can be used as a data source in MicroStrategy. You can use Data Import to quickly include data from text files in your MicroStrategy project. Steps to use Data Import to import data and begin your analysis is included in the *MicroStrategy Web Help*.

Creating Database Connections in Web

In MicroStrategy Web, users can import data from different data sources, such as a database or the results of a Freeform query, then create reports, documents, and dossiers to report on their imported data. You can define a new database connection directly from Web for users to import data from, or edit, delete, rename, or duplicate an existing connection.

You must have the Create and Edit Database Instances and Connections and Create and Edit Database Logins privileges to define a new database connection.

If you plan to connect to a data source using a DSN, the DSN must be created and available. If a DSN is not available, you can use the DSNLess Connection option to connect to your data source.

To Create a New Database Connection

1. In MicroStrategy Web, navigate to any folder page, such as Shared Reports or My Reports.
2. From the navigation bar on the left, go to **Create > Access External Data > Database**. The Select Import Options dialog box opens.

3. Select **Pick Tables** to select single or multiple tables to import data from.
   - If you want to use a graphical interface to build the SQL query to use to import your data, select **Build a Query**.
   - If you want to manually type or paste a query to import your data, select **Type a Query**.

4. Click **OK**.

5. From the **Data Sources** panel, click **Add**.

6. Select the type of connection to your database, as follows:
   - To connect to a data source using a DSN, select **DSN Connections**. Select the DSN of the database that you want to connect to from the DSN drop-down list, then select the appropriate database management system (DBMS) from the DBMS drop-down list.
   - To connect directly to a data source, select **DSNless Connections**.
   - If you clear the **Show databases whose drivers were not found** check box, only databases that have an installed and configured driver are available for selection. These databases can be connected to by selecting the required **Database** and **Version** from the drop-down lists, and supplying the required connection information. For a detailed list of the information required for each database type, see *Creating DSNs for Specific Data Sources, page 759.*
• If you select the **Show databases whose drivers were not found** check box, additional databases that do not have a configured driver are available for selection. These databases can be connected to by selecting the required **Database** and **Version** from the drop-down lists, and then configuring a connection to the database by completing the following steps:

1. Click **Show connection string**.

2. Type the value for each configuration requirement listed. Depending on the database you are connecting to, this includes the server name, port number, and database name. For a detailed list of the information required for each database type, see *Creating DSNs for Specific Data Sources, page 759*.

3. Select the **Edit connection string** check box.

4. Modify the **Driver={DriverName}** part of the connection string, where **DriverName** is the default name used for the driver. Replace the default **DriverName** with the name of the driver that your administrator installed for the database.

5. If there were any optional configuration parameters that you chose not to define, modify the connection string to remove the parameters completely from the string. These parameters are listed with an equal sign (=) followed immediately by a semicolon (;), indicating no value is provided. For example, if the connection string includes **AlternateServers=;** remove this text from the connection string.

6. Type a user name and password with access to the database in the **User** and **Password** fields.
8. Type a name for the database connection in the **Data Source Name** field.

9. Do one of the following:
   - To allow other users to import data using the database connection, select the **Share this connection with everybody** check box.
   - To deny other users the ability to import data using the database connection, clear the **Share this connection with everybody** check box.

10. Click **OK**.

**Configuring ODBC Parameters with ODBC.ini**

The `odbc.ini` file is the configuration file that stores the definitions for all the ODBC DSNs in a Linux environment. Therefore this section is not relevant to ODBC and DSN connections on Windows.

For information on what drivers to use with MicroStrategy, see the [Drivers and Connectors](#) page.

These ODBC DSNs are defined by specifying values for certain DSN parameters. This file is activated by the environment variable `ODBCINI`, and is required by all ODBC applications. By default, the `odbc.ini` file is installed in `HOME_PATH`, where `HOME_PATH` is the directory you specified as the home directory during installation on Linux. It contains the definitions for all the MicroStrategy-branded ODBC drivers.

MicroStrategy supports ODBC drivers from other vendors that you can install separately. This involves manually defining the DSN parameters in the `odbc.ini` file.
Modification of the `odbc.ini` file is necessary to configure the full list of ODBC driver settings or for ODBC drivers that are not accessible through the MicroStrategy Connectivity Wizard. However, caution should be taken when modifying the `odbc.ini` file as incorrect modifications can cause unintended functionality and errors.

Refer to the `odbc.ini.example` file installed in `HOME_PATH`, where `HOME_PATH` is the directory you specified as the home directory during installation on Linux. It is recommended to copy the examples in the `odbc.ini.example` file to `odbc.ini`, to act as a basis for your configurations. This example file provides commonly used settings for the driver parameters.

If you require additional information on the purpose of and available options for each parameter, refer to the vendor documentation.
MicroStrategy Hadoop Gateway
Simplified Installation

Introduction to the MicroStrategy Hadoop Gateway ........................................ 804
Environment Considerations .................................................................................. 806
Using the MicroStrategy Hadoop Gateway .......................................................... 813
Troubleshooting ..................................................................................................... 822
Known Issues .......................................................................................................... 831
Frequently Asked Questions .................................................................................... 833

Introduction to the MicroStrategy Hadoop Gateway

Assumptions

- This document is only eligible for MicroStrategy MicroStrategy Hadoop Gateway 10.11 and later versions. Previous versions are not fully covered.

- The Intelligence Server and MicroStrategy Hadoop Gateway version numbers must be identical. Otherwise, data files may not be imported successfully.
• The simplified installation process of MicroStrategy MicroStrategy Hadoop Gateway is certified in Cloudera CDH 5.10, 5.11, 5.12 and 5.13; Hortonworks 2.4, 2.5, and 2.6. MicroStrategy Hadoop Gateway does not support MapR and other clusters.

• Starting with MicroStrategy 10.11, MicroStrategy Hadoop Gateway will be launched with spark-submit.

The application detects Hadoop and Java dependencies and resolves them automatically. MicroStrategy also provides automatic deployment via MicroStrategy Hadoop Gateway Manager. Refer to MicroStrategy Community for more details.

Introduction

MicroStrategy Hadoop Gateway is a new data processing engine, introduced in MicroStrategy 10.6, that can be installed in a Spark environment. This native connector allows analysis of data in Hadoop and provides high-speed parallel data transfer between the Hadoop Distributed File System (HDFS) and the MicroStrategy Intelligence Server.

MicroStrategy Hadoop Gateway is a native connector which was built based on Spark 1.6.x. You can choose one or more data files from Hadoop HDFS and load them into MicroStrategy Intelligence Server. Data files can be published as either an In-Memory Cube or a Live Connect Cube.

MicroStrategy Hadoop Gateway supports three operation modes:

• YARN client

• Standalone (HDP cluster does not support Standalone)

• Local

In YARN client mode, the driver runs in the client process, and the application master is only used for requesting resources from YARN.
Running in this mode requires YARN service to be enabled on your Hadoop cluster.

In Standalone mode, MicroStrategy Hadoop Gateway uses Spark Master to coordinate data processing.

In Local mode, no application is deployed in the Spark nodes. All data processing is executed in the MicroStrategy Hadoop Gateway host. This modality is intended for testing and troubleshooting.

Related Topics

Environment Considerations

How to Deploy the MicroStrategy Hadoop Gateway

How to Start the MicroStrategy Hadoop Gateway

How to Register the MicroStrategy Hadoop Gateway

Troubleshooting

Known Issues

Frequently Asked Questions

Environment Considerations

Security on Data Access (Authentication)

Accessing your Cluster services may be controlled by a compliant Kerberos implementation (Kerberos MIT, Active Directory). In a Kerberos environment, the MicroStrategy Hadoop Gateway can identify itself as a Kerberos principal and have access to the required services: HDFS, Spark Manager.
Hadoop as an Edge or Proxy Mode of the Cluster

We recommend the MicroStrategy Hadoop Gateway host to be part of the Hadoop cluster for security, administration and performance benefit. An Edge or Proxy node is physically or logically located within the cluster, and contains the same set of libraries.

From an administration point of view, any upgrade on the cluster library version will include the Edge or Proxy node. It benefits performance, as the data transfer speed should be higher. It improves security, as the node could be constrained to the same rules and authentication.

High Availability Mode in HDFS and YARN Cluster Services

Following best practices, the cluster may have implemented High Availability (HA) mode on the services. A server node can be set as Active and enabled while an additional can be set to Standby to replace the Active at any time. An HA environment would use a different set of properties when referring to these services. Review your environment and make sure it runs in HA mode.

System Requirements and Supported Configurations

The system requirements for a MicroStrategy Hadoop Gateway are the same as for a Spark cluster. The supported Spark version is 1.6.x. Supported distribution version for MicroStrategy Hadoop Gateway is Cloudera Data Hub 5.10 or above, and Hortonworks 2.4 or above.

For Cluster environments with a standard authentication mechanism, the MicroStrategy Hadoop Gateway can be operated in Local, YARN client and Spark Standalone mode. For environments with Kerberos authentication enabled, the MicroStrategy Hadoop Gateway can only be operated in YARN client mode.

The following are needed on a Hadoop cluster:
• You should have a Hadoop environment installed on Unix/Linux servers.

The Hadoop Cluster must have at least the HDFS service installed. Other services that can be installed include Hive, Hue, Oozie, and ZooKeeper.

• MicroStrategy Hadoop Gateway supports the High Availability mode of NameNode and YARN Resource Manager.

To enable NameNode High Availability, see *How to Browse the Hadoop Distributed File System and Preview Files*. No extra configuration is required for YARN Resource Manager High Availability. It is handled automatically.

• If you are using the MicroStrategy Hadoop Gateway in YARN client mode, the Hadoop Cluster should have YARN and Spark services installed.

• If you are using MicroStrategy Hadoop Gateway on Spark Standalone mode, the Hadoop cluster should have Spark (Standalone) service installed.

• Connectivity parameters to the Spark master (for example `spark://SparkMasterNode:7077`)

• Cloudera Manager does not allow this service to be installed if the cluster has Kerberos enabled.

• For troubleshooting purposes:

  • Access to Spark Standalone website
    `http://SparkMasterNode:18080`

  • Access to Spark History Server website
    `http://SparkHistoryServerNode:18088`
- Access to YARN resource monitor website
  http://YARNResourceManagerNode:8088

- Make sure Spark service is installed and configured properly. MicroStrategy Hadoop Gateway 10.11 and later will be launched with spark-submit methodology to avoid issues with cluster environment compliance and compatibility.

## Ports Used by MicroStrategy Hadoop Gateway

<table>
<thead>
<tr>
<th>From</th>
<th>To: Service Default Port</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligence Server</td>
<td>MicroStrategy Hadoop Gateway Host Port 30004</td>
<td>Sending commands from the Intelligence Server to MicroStrategy Hadoop Gateway to fetch data. The port number is configurable in MicroStrategy Hadoop Gateway configuration file: /conf/hgos-spark.properties.</td>
</tr>
<tr>
<td>Intelligence Server</td>
<td>MicroStrategy Hadoop Gateway Host Port 4020</td>
<td>Port used by Intelligence Server to browse HDFS via MicroStrategy Hadoop Gateway RESTful service. The port number is configurable in MicroStrategy Hadoop Gateway configuration file: /conf/hgos-spark.properties.</td>
</tr>
<tr>
<td>MicroStrategy Hadoop Gateway</td>
<td>HDFS NameNode Port 8020</td>
<td>Default port number is: 8020. Please contact your cluster administrators for specific port number.</td>
</tr>
<tr>
<td>HDFS (all nodes of cluster Hadoop)</td>
<td>Intelligence Server Port 30241</td>
<td>Used to send query result set from MicroStrategy Hadoop Gateway Spark application worker nodes to Intelligence Server. The port number is configurable in the OS registry where the Intelligence Server is installed. Registry key: HKEY_LOCAL_MACHINE/SOFTWARE/Wow6432Node/MicroStrategy</td>
</tr>
</tbody>
</table>
From | To: Service Default Port | Explanation
--- | --- | ---
| | gy/DSS Server/Castor/DSPort Registry file in Linux: MSIReg.reg
MicroStrategy Hadoop Gateway | YARN Resource Manager Port 8032 | YARN connectivity
MicroStrategy Hadoop Gateway | Spark Port 4040 | Spark connectivity
MicroStrategy Hadoop Gateway | Kerberos KDC Port 88 | To authenticate MicroStrategy Hadoop Gateway to access other services (such as HDFS).

If Kerberos Authentication Has Been Enabled

To learn about Kerberos installation, see: *How to Install the Kerberos Authentication Service*.

Please refer to following links about how to enable Kerberos authentication in Cloudera CDH and Hortonworks HDP cluster.

You will need a Kerberos principal (or SPN in Active Directory) to authenticate your MicroStrategy Hadoop Gateway process.

The Kerberos authentication happens in at least two events:

- Browsing the HDFS file directory to select files to import.
- MicroStrategy Hadoop Gateway directly connects NameNode.
Starting the MicroStrategy Hadoop Gateway on YARN-client mode:

MicroStrategy Hadoop Gateway will deploy Spark applications across YARN and requires a Kerberos ticket for this.

MicroStrategy Hadoop Gateway should be executed with a valid Linux user account linked to a Kerberos principal. It could have any name, but for convention we will refer to it as

```
hgos/<HadoopGatewayHostFQDN>REALM_NAME
```

As any other Cluster account, this account should be able to log into all machines of the cluster.

This account should be allowed to log into HDFS with write privileges in its home directory (for example

```
hdfs://NameNode:8020/user/hgos
```

Cluster nodes should have required libraries to work as a Kerberos client (these could be the packages `krb5-workstation`, `openldap-client`).

If High Availability Mode Has Been Enabled

Identify the nameservice of the HDFS service.

The following are needed on a MicroStrategy Hadoop Gateway driver machine:

- Host OS: Linux (recommended: CentOS-7).
- The host to be part of the CDH cluster as a proxy node or worker node.
- Java Runtime Environment version 1.7 or 1.8 (latest subversion available) installed.
- Linux account must have write and execute privileges in the installation folder.
The OS account should have an assigned user folder in HDFS and read/write privileges (for example
hdfs://<HDFSNameNode:8020>/user/<Principal name>/) (a temp directory .sparkStaging will be created).

Connectivity parameters, IP address and the port to connect from Intelligence Server.

For detailed logs, replace the log4j.properties file with the richer version available in the troubleshooting section.

If Kerberos Authentication Has Been Enabled

The host should have installed Kerberos client libraries (like krb5-workstation) and allow Kerberos commands like kinit or klist.

The Java Runtime Environment should have the Java Cryptography Extension libraries to support aes-256 encryption. There libraries are available at Oracle’s website. The Java JCE package contains two JAR libraries. Use these and replace them in the directory <JRE_HOME>/lib/security (If JDK is installed instead of JRE, it should be <JDK_Home>/jre/lib/security. Keep a backup of your original libraries).

For the Intelligence Server host, update firewall and network rules to allow connectivity into port 30241 from cluster worker nodes.

Related Topics

Introduction to the MicroStrategy Hadoop Gateway
How to Deploy the MicroStrategy Hadoop Gateway
How to Start the MicroStrategy Hadoop Gateway
How to Register the MicroStrategy Hadoop Gateway
Troubleshooting

Known Issues

Frequently Asked Questions

Using the MicroStrategy Hadoop Gateway

Using MicroStrategy Hadoop Gateway requires you to complete three steps: deploying, starting, and registering data.

- **How to Deploy the MicroStrategy Hadoop Gateway**

  Deploying the MicroStrategy Hadoop Gateway on the host machine requires transferring, extracting, and configuring the MicroStrategy Hadoop Gateway.

- **How to Start the MicroStrategy Hadoop Gateway**

  Starting the MicroStrategy Hadoop Gateway will start the MicroStrategy Hadoop Gateway service and will be ready to accept commands from the Intelligence Server (through specified TCP port).

- **How to Register the MicroStrategy Hadoop Gateway**

  Registering the MicroStrategy Hadoop Gateway stores the configuration of the MicroStrategy Hadoop Gateway in the Intelligence Server environment. In this stage, you define the operation mode and connectivity parameters.

How to Deploy the MicroStrategy Hadoop Gateway

Deploying the MicroStrategy Hadoop Gateway on a host machine requires to transfer, extract and configure MicroStrategy Hadoop Gateway.
How to Obtain MicroStrategy Hadoop Gateway

You will need a copy of the MicroStrategy Hadoop Gateway package. It can be obtained from MicroStrategy download website or from Intelligence Server directory.

To Download the MicroStrategy Hadoop Gateway

2. Click on Feature Releases and select your version (for example 10.x).
3. Click on MicroStrategy 10.x Add-Ons and Tools.
4. Click to download Hadoop Gateway.
5. Click OK
6. Save the file 10.x.00xx.00xx_10.x_GA_HadoopGateway.zip.
7. Copy the file onto your MicroStrategy Hadoop Gateway machine.

To Copy the MicroStrategy Hadoop Gateway From the Intelligence Server

1. Go to the Intelligence Server directory.
2. Open the "hgos-manager" directory.

How to Install the MicroStrategy Hadoop Gateway

1. Log in to the MicroStrategy Hadoop Gateway host machine via SSH.
2. Locate the MicroStrategy Hadoop Gateway tarball `hgos-dist.tar.gz`.

3. Unzip it in your selected directory (for example, by running: `tar -xvzf hgos-dist.tar.gz`).

How to Configure MicroStrategy Hadoop Gateway

MicroStrategy Hadoop Gateway will be configured with minimum infrastructure after installation.

To customize the installation to specific requirements, modify the MicroStrategy Hadoop Gateway configuration file `<Installation path>/conf/hgos-spark.properties`.

If Kerberos Authentication Has Been Enabled

Modify MicroStrategy Hadoop Gateway configuration file `<MicroStrategy Hadoop Gateway installation path>/conf/hgos-spark.properties` and uncomment the following section:

```
spark.yarn.token.renewal.interval=50000
spark.yarn.security.tokens.hdfs.enabled=true
spark.yarn.principal=<Kerberos Principal>
spark.yarn.keytab=<Path of principal keytab file>
```

Keep at hand the MicroStrategy Hadoop Gateway host IP, host name, and connectivity parameters.

Related Topics

- Introduction to the MicroStrategy Hadoop Gateway
- Environment Considerations
- How to Deploy the MicroStrategy Hadoop Gateway
How to Register the MicroStrategy Hadoop Gateway

Troubleshooting

Known Issues

Frequently Asked Questions

How to Start the MicroStrategy Hadoop Gateway

Starting MicroStrategy Hadoop Gateway will start the MicroStrategy Hadoop Gateway service and will be ready to accept commands from the Intelligence Server (through specified TCP port).

Launch MicroStrategy Hadoop Gateway Service

1. Run the file /<MicroStrategy Hadoop Gateway installation path>/sbin/start-hgos.sh.

2. Once the MicroStrategy Hadoop Gateway is started successfully, the result should look like below:

```
SPARK_HOME is ''
[PASS] jdk version 1.8 is OK
[PASS] Hadoop 2.6 is OK
[PASS] cdh 5.13 is OK
[PASS] Spark version is 1.6
[FAIL] WARN: No kerberos config found. If kerberos is not enabled, please just ignore this message, otherwise HGOS will fail to start

###############################################################################
##########
HGOS use spark setting from hgos-spark.properties, spark master is yarn-client
HGOS will run using yarn-client mode, according to your hgos-spark.properties.
==================Starting
HGOS==================
pid: 9089, starting, and more detail is in the log file
```
Related Topics

Introduction to the MicroStrategy Hadoop Gateway

Environment Considerations

How to Deploy the MicroStrategy Hadoop Gateway

How to Register the MicroStrategy Hadoop Gateway

Troubleshooting

Known Issues

Frequently Asked Questions

How to Register the MicroStrategy Hadoop Gateway

Registering MicroStrategy Hadoop Gateway stores the configuration of the MicroStrategy Hadoop Gateway in the Intelligence Server environment. In this stage, you define the operation mode and connectivity parameters.

How to Browse the Hadoop Distributed File System and Preview Files


2. Click Create > Add External Data.
3. In the Connect your Data dialog, click **Hadoop**.

4. Select **Browse Hadoop Files using MicroStrategy Big Data Engine** when "Select Import Options" form appears.

   - **Browse Hadoop Files using MicroStrategy Big Data Engine**
     
     Import tables by browsing HDFS (Hadoop Distributed File System)

   - **Build a Query**
     
     Use a graphical interface to build a 'SQL on Hadoop' query to import a table

   - **Type a Query**
     
     Paste or write a query script to import a table via 'SQL on Hadoop' or Pig

   - **Select Tables**
     
     Select single or multiple tables to import via 'SQL on Hadoop'
5. Click **Next**. The "Connect to Hadoop" form appears with the HDFS file browser.

6. In Hadoop File Browser entry, click **Change Connection** on top of screen.

7. Select **Edit connection string** to enable connection string modification.
8. Make sure you configure appropriate Hadoop NameNode IP/Host and port in the connection string.

Hadoop NameNode IP address is: 10.242.109.2, and HDFS port number is by default: 8020. The MicroStrategy Hadoop Gateway is deployed on host IP: 10.242.109.10 with port: 30005 (port can be configured in MicroStrategy Hadoop Gateway configuration file). The connection string for the above data should look like this:

```
hadoopName=10.242.109.2;hdfsPort=8020;BDEIP=10.242.109.10;BDEPORT=30005
```
If Hadoop NameNode High Availability Has Been Enabled

1. Ask your Hadoop cluster administrators about the HDFS nameservice.

2. Replace the attribute `hadoopName` with `hadoopNameService` in the connection string, and remove the attribute `hdfsPort`.

   Here, the HDFS nameservice is "nameservice1" in the cluster. The connection string should look like this:

   ```
hadoopNameService=nameservice1;BDEIP=10.242.109.10;BDEPORT=30005;
   ```

   All of attributes in the connection string are case sensitive.

3. Click **OK** to save connectivity parameters and finish MicroStrategy Hadoop Gateway registration. HDFS folders and files will appear if everything has been configured properly.

Once the MicroStrategy Hadoop Gateway is running, the Import Data feature can be used to import data as an Intelligent Cube for data consumption.

Related Topics

*Introduction to the MicroStrategy Hadoop Gateway*

*Environment Considerations*

*How to Deploy the MicroStrategy Hadoop Gateway*

*How to Start the MicroStrategy Hadoop Gateway*

*Troubleshooting*

*Known Issues*

*Frequently Asked Questions*
Troubleshooting

How to Get Additional Log Messages to Troubleshoot the MicroStrategy Hadoop Gateway


2. Update its contents to match the lines below:

```plaintext
log4j.rootLogger=DEBUG,console

# To enable logging for all classes of package com.microstrategy and its subpackages, specify the logger as shown. Or you can specify any name for the logger.
# For example, you can name your logger MICROLOG as shown in the following statement:
log4j.logger.com.microstrategy=INFO, MICROLOG

log4j.logger.org=WARN
```

How to Get YARN Log Messages to Troubleshoot the MicroStrategy Hadoop Gateway

If You Are Using the Cloudera CDH Cluster

1. Log into Cloudera Manager, the cluster administration portal.

2. Click YARN service.

3. Click Web UI.

4. Select the active ResourceManager WebUI if YARN Resource Manager is configured as High Availability. You will be
redirected to the Hadoop Resource Manager page.
If You Are Using the Hortonworks Cluster

1. Log into Ambari, the cluster administration portal.

2. Click **YARN service**.

3. Click **Quick Links**.

4. Select **ResourceManager UI**. You will be redirected to the Hadoop Resource Manager page.
5. Find Hadoop Gateway in the application list, and click the application ID.

You will find application details and logs in YARN.
What to Do if You Get an Error in the MicroStrategy Hadoop Gateway Connection Page and You Are Unable to Browse HDFS

Follow the steps below to diagnose the issue.

1. Make sure MicroStrategy Hadoop Gateway service has been started successfully and port number 4020 is open on
MicroStrategy Hadoop Gateway host machine. In MicroStrategy2019, the port 4020 is configurable in Hadoop. You can change the value (port number) of parameter hgos.restful.port in <MicroStrategy Hadoop Gateway installation path>/conf/hgos-spark.properties.

2. Check if the HDFS service is running and NameNode is accessible from the MicroStrategy Hadoop Gateway host machine.

3. Check if an appropriate connection string is configured and the values are correct. All the connection string attributes are case sensitive.

Connection string template:

```
hadoopName=<Active Hadoop NameNode>;hdfsPort=<NameNode port>;BDEIP=
```

If Hadoop NameNode High Availability Is Configured

1. Ask your cluster administrators about the HDFS nameservice.

2. Replace the attribute hadoopName with hadoopNameService in the connection string, and remove the attribute hdfsPort.

```
hadoopNameService=<HDFS nameservice>;BDEIP=<MicroStrategy Hadoop Ga
```

If Kerberos Authentication Has Been Enabled

1. Make sure the principal ticket is valid. You can refresh the ticket by running the following command:

```
kinit -k -t <Principal keytab file path> <Principal> -c <Cache name>
```

2. Make sure that the present principal has sufficient permissions to browse HDFS. You can do so by running the following command:
The expected result is like the output below:

```
[root@ash-109-32r sbin]# hadoop fs -ls /
Found 11 items
-rw-r--r--  3 hdfs  supergroup  1996573 2017-01-10 03:09 /0_0_0_parquet
drwxr-xr-x  hdfs  supergroup          0 2017-05-23 08:43 /Blitz
-rw-r--r--  3 root   supergroup        21 2017-06-26 03:26 /DE65820.txt
drwxrwxrwx  root   supergroup          0 2017-02-09 06:16 /DataSets
-rw-r--r--  3 root   supergroup        189 2017-11-13 06:49 /Date_Format.csv
-rw-r--r--  3 root   supergroup        308 2017-11-08 07:31 /Date_Format_b.csv
-rw-r--r--  3 root   supergroup        476 2017-11-01 10:26 /Date_Format_c.csv
drwxr-xr-x  root   supergroup          0 2017-08-31 12:53 /Demo_Files
drwxrwxrwx  root   supergroup          0 2017-07-13 14:56 /EPA_3C
drwxr-xr-x  root   supergroup          0 2017-09-07 02:55 /EPA_3C_Part
```

In-memory cubes cannot be published or generate dossiers in the Live Connect mode. By using the MicroStrategy Hadoop Gateway log, you can determine if data is transmitted back to Intelligence Server successfully from the Hadoop cluster.

Make sure that the port number used to receive data from MicroStrategy Hadoop Gateway is open on the Intelligence Server. By default, the port number is 30241, and it is configurable in the registry.
If the intelligence Server and Hadoop cluster are installed on separate networks, the Intelligence Server communicates with the cluster via a gateway. You need to specify the Intelligence Server host name and IP. Otherwise, the MicroStrategy Hadoop Gateway will submit data to the gateway machine instead, because the MicroStrategy Hadoop Gateway detects the source automatically from the network request.

To specify the Intelligence Server host name and IP, you need to modify the registry key and manually set `DSHostName` with the Intelligence Server host name and IP.

The ability to specify the Intelligence Server hostname and IP is only available for MicroStrategy 10.10 and later.

Registry key:

```
HKEY_LOCAL_MACHINE/SOFTWARE/Wow6432Node/MicroStrategy/DSS Server/Castor/DSHostName
```

Registry file on the Intelligence Server:

```
MSIReg.reg
```


Related Topics

- Introduction to the MicroStrategy Hadoop Gateway
- Environment Considerations
- How to Deploy the MicroStrategy Hadoop Gateway
- How to Start the MicroStrategy Hadoop Gateway
- How to Register the MicroStrategy Hadoop Gateway
- Known Issues
Frequently Asked Questions

Known Issues

Symptom

In MicroStrategy Web, users have an expansion in the variety of data sources to which they can connect. These new options are under Add External Data. A user with Web Reporter, Web Analyst, and Web Professional privileges will receive a "You do not have enough privileges to perform this operation" error when trying to import a table from a data source. This is due to the fact that the user privilege "Access data from Databases, Google BigQuery, BigData, OLAP, BI tools" is not the only privilege needed. Creating a data source object to import Tables is considered an Administrator Privilege.

Workaround

Give the user the privilege from the Administration group "Create and edit database and instances and connections".

<table>
<thead>
<tr>
<th>User Definition - Project Access</th>
<th>Human Resources Analyst</th>
<th>MicroStrategy Tutorial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create and edit database and instances and connections</td>
<td>☑</td>
<td>☑</td>
</tr>
</tbody>
</table>

The "Create and edit database and instances and connections" is a Server level only privilege and can only be granted only at the project source level, so it needs to be given at a user or user group level.
Symptom

There are persisting tables in the Hive Metastore when the MicroStrategy Hadoop Gateway service starts. In the log file you can see that the MicroStrategy Hadoop Gateway service is trying to modify the structure of some Metastore tables (such as DBS tables). When it tries to commit the transaction, an error is produced and a rollback operation is generated.

Workaroud

MicroStrategy Hadoop Gateway requires a Hive component to load ORC files, and basic metadata needs to be created to establish the Hive component. Modifying the structure of Metastore tables is not permitted by default.

To enable Metastore modification, add the following section to `/etc/hive/conf/hive-site.xml`.

```xml
<property>
  <name>datanucleus.autoCreateSchema</name>
  <value>true</value>
</property>
```

Alternatively, the Hive component can be disabled if you are not going to import ORC files. Set the property `hgos.spark.enable` value to `false` in the MicroStrategy Hadoop Gateway configuration file `<MicroStrategy Hadoop Gateway installation path>/conf/hgos-spark.properties`.

Symptom

Text columns in the parquet file created by Impala are loaded as binaries.

Workaroud

The Impala Parquet writer always creates an unannotated binary field
for string columns. It is possible to specify that the field include the UTF-8 annotation to indicate UTF-8 string data. It was resolved in Impala 2.6 and CDH 5.11 has 2.8.

If all these columns need to be converted to UTF-8 strings, add `spark.sql.parquet.binaryAsString=true` into configuration file `<MicroStrategy Hadoop Gateway installation path>/conf/hgos-spark.properties`.

Related Topics

*Introduction to the MicroStrategy Hadoop Gateway*

*Environment Considerations*

*How to Deploy the MicroStrategy Hadoop Gateway*

*How to Start the MicroStrategy Hadoop Gateway*

*How to Register the MicroStrategy Hadoop Gateway*

*Troubleshooting*

*Frequently Asked Questions*

**Frequently Asked Questions**

*What are the recommended settings for `hgos-spark.properties`?*

The table below helps calculate the recommended settings based on the number of working nodes, RAM and virtual cores in each node, and the number of executors to be allocated on each worker node.
### Recommended performance parameters for YARN client mode

<table>
<thead>
<tr>
<th>ID</th>
<th>Item</th>
<th>Parameter</th>
<th>Formula</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Number of Node</td>
<td></td>
<td></td>
<td>2</td>
<td>Available in your hardware</td>
</tr>
<tr>
<td>C2</td>
<td>RAM per Node (GB)</td>
<td></td>
<td></td>
<td>380</td>
<td>Available in your hardware</td>
</tr>
<tr>
<td>C3</td>
<td>VCPUs per Node</td>
<td></td>
<td></td>
<td>40</td>
<td>Available in your hardware</td>
</tr>
<tr>
<td>C4</td>
<td>Total number of VCPUs</td>
<td></td>
<td>( C_1 \times C_3 )</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>Allocated executors</td>
<td></td>
<td>( S_1 = S_2 \times C_1 )</td>
<td>48</td>
<td>Number of executors to be allocated on each worker node</td>
</tr>
<tr>
<td>S2</td>
<td>Executors per Node</td>
<td>spark.executor.cores</td>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>Max memory per executor (GB)</td>
<td>( S_3 = \frac{C_2}{S_2} )</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H1</td>
<td>Overhead (GB)</td>
<td></td>
<td>( H_1 = S_3 \times 0.07 )</td>
<td>4</td>
<td>Overhead memory used by the OS. It defaults to 0.07 ( \times ) spark.executor.memory</td>
</tr>
</tbody>
</table>
### Installation and Configuration Guide

<table>
<thead>
<tr>
<th>ID</th>
<th>Item</th>
<th>Parameter</th>
<th>Formula</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H 2</td>
<td>Number of executors</td>
<td>spark.executor.instances</td>
<td>H2 = S1 - 1</td>
<td>47</td>
<td>Total number of executors created in the cluster. One node needs to occupy an executor for AM.</td>
</tr>
<tr>
<td>H 3</td>
<td>Memory per executor (GB)</td>
<td>spark.executor.memory</td>
<td>H3 = S3 - H1</td>
<td>59</td>
<td>Running executors with too much memory often results in excessive garbage collection delays. 64 GB is a rough guess at a good upper limit for a single executor.</td>
</tr>
<tr>
<td>H 4</td>
<td>Cores per executor</td>
<td></td>
<td>H4 = (C3 / S2) - 1</td>
<td>6</td>
<td>Leave 1 core for system processes</td>
</tr>
</tbody>
</table>

**Can MicroStrategy Hadoop Gateway release cluster resources during idle time?**

Yes, MicroStrategy Hadoop Gateway supports releasing cluster resources while the service is idle. You will have to configure the MicroStrategy Hadoop Gateway executors and cores as dynamically allocated to enable this behavior. Modify the MicroStrategy Hadoop Gateway configuration file `<MicroStrategy Hadoop Gateway installation path>/conf/hgos-spark.properties`, and uncomment the dynamic allocation section.

**Does MicroStrategy Hadoop Gateway support Live Connect?**
Yes, MicroStrategy Hadoop Gateway supports Live Connect Cube since MicroStrategy 10.9. There is no extra configuration required to enable it.

What is the minimum memory and hard drive space to install MicroStrategy Hadoop Gateway on the host machine?

The minimum requirement for the MicroStrategy Hadoop Gateway is 256 MB of disk space and 2 GB of memory.

What is the memory and hard drive space necessary in other nodes (NameNode, DataNode) while MicroStrategy Hadoop Gateway is running?

MicroStrategy Hadoop Gateway will not start any extra processes on a NameNode or a DataNode. MicroStrategy Hadoop Gateway just submits job to Spark.

DataNode memory usage depends on the number set by the customer to attribute file spark.executor.memory in the configuration file (by default, it is 1 GB). Meanwhile, NameNode memory usage will not be significantly affected.

Will MicroStrategy Hadoop Gateway create any metadata on HDFS?

When MicroStrategy Hadoop Gateway starts, some JAR files will be uploaded to HDFS to hdfs://HDFSNameNode:8020/user/${user_name_start_hgos}/.sparkStaging. By default, the files in sparkStaging will be deleted automatically once MicroStrategy Hadoop Gateway service is shut down.

Total size of JAR files will be no larger than 256 MB.

Can MicroStrategy Hadoop Gateway refresh Kerberos principal ticket automatically?

No, MicroStrategy Hadoop Gateway will not refresh Kerberos principal ticket automatically. You will have to refresh it by running
the `kinit` command or creating a cron job to refresh the ticket on schedule.

A template for `kinit_cron.sh`:

```bash
function setup_kerberos() {
    echo "klist:"
    klist

echo "KRB5CCNAME env:"
    export KRB5CCNAME="HGOS_HOME/conf/krb5cc_hgos"
    echo $KRB5CCNAME

echo "kinit"
    kinit -kt $keytab_path $principal_name -l 1d5h -r 2d -f

    echo "klist"
    klist
}
setup_kerberos
```

To schedule a cron job, run:

```
root@HOST # crontab -l 0 */2 * * * <path to file>/kinit_cron.sh
```

**Does MicroStrategy Hadoop Gateway support Apache Sentry?**

Yes, MicroStrategy Hadoop Gateway supports HDFS ACL by Apache Sentry, and no extra configuration is required. See the video below for how MicroStrategy Hadoop Gateway works with Apache Sentry.
Does MicroStrategy Hadoop Gateway support Apache Spark 2 and later?

No. MicroStrategy Hadoop Gateway is built on Spark 1.6 and only eligible to be deployed in a Spark 1.6 environment. We are working to release MicroStrategy Hadoop Gateway on Spark 2.

Related Topics

Introduction to the MicroStrategy Hadoop Gateway

Environment Considerations

How to Deploy the MicroStrategy Hadoop Gateway

How to Start the MicroStrategy Hadoop Gateway

How to Register the MicroStrategy Hadoop Gateway

Troubleshooting

Known Issues
Installing and Configuring Topology

The new MicroStrategy Workstation Topology feature provides users with a graphical interface that presents the entire topology of every node within their environment, in addition to related services for available nodes. Through the Topology menu in Workstation, administrators can monitor the current state of every service on each node, as well as start or stop individual services.

This section describes how to install and configure topology.
Introduction to Topology

Topology allows administrators to monitor MicroStrategy Services and manage them in MicroStrategy Workstation. The following functionality is supported:

- **Service Health Monitor**
  
  The administrator can monitor MicroStrategy services through Workstation and visualize if they are running or stopped.

- **Service Start/Stop**
  
  The administrator can start or stop MicroStrategy services through Workstation. This functionality is currently limited to SSH authentication with username and password.

See the topic, *How to View Environment Topology and Monitor Services* in the *Workstation Online Help* for more information about using topology within Workstation.

The two following types of lightweight agents are used to monitor services in Workstation:

- **Monitoring Agents**
  
  The monitoring agents come with the installation of most MicroStrategy services. Use them to monitor services and view their health status.

- **Communication Agents:**
  
  Some monitoring agents also act as communication agents. In addition to monitoring services, communication agents help other monitoring agents locate each other and gather monitoring information. The administrator must pick at least one agent to act as a communication agent, so all other agents can locate each other through this agent.
On environments where three or more machines host MicroStrategy Services, it is recommended that you choose at least three communication agents to provide redundancy and improved reliability, in case one communication agent becomes unavailable. The machines chosen to act as communication agents must be machines that host MicroStrategy Services, such as Intelligence Server or MicroStrategy Library. If you decide to use only one communication agent, it is recommended that you choose the machine that houses MicroStrategy Library (if deployed through the MicroStrategy installation), since Library communicates directly with Workstation.

If your Library deployment is done through a WAR file or if you do not use a machine that houses MicroStrategy Library to host the communication agent, then use a machine that houses Intelligence Server.

MicroStrategy uses **Consul** technology for Services Registration. A monitoring agent corresponds to a Consul agent in client mode. A communication agent corresponds to a Consul agent in server mode.
Installing Topology

This section describes how to install topology using the Topology Configuration dialog of the Installation wizard.

1. On the Topology Configuration dialog of the Installation wizard, select whether your environment has **Single** or **Multiple** machines.

2. If your environment contains multiple machines, determine which
machines in your environment you want to act as communication agents. Among all the machines in your environment that will host MicroStrategy Services, select at least one of them (three are recommended for multiple machine environments) to act as a communication agent. Enter each machine in the text field, separated by semicolons. You must input the exact same list on all machines, using the Topology Configuration dialog of the Installation wizard, including the monitoring and communication agent machines.

The MicroStrategy Services are as follows:

- MicroStrategy Intelligence Server
- MicroStrategy Web Universal
- MicroStrategy Library
- MicroStrategy Mobile
- MicroStrategy Messaging Services
- MicroStrategy Platform Analytics
- MicroStrategy Certificate Store
- MicroStrategy Identity

An odd number of communication agents is required due to the leadership selection algorithm.

Examples:

```
servername1.domain.com;servername2.domain.com;
servername3.domain.com;
servername1.domain.com;
```

Example: Your environment contains the following two machines and you select one machine to host the communication agent:
• Machine 1 hosts MicroStrategy Library and the communication agent

• Machine 2 hosts the Intelligence Server

When performing the installation, enter the full domain name, machine1.domain.com, on both machines.

Example: Your environment contains the following three machines and you want all machines to host a communication agent:

• Machine 1 hosts MicroStrategy Library

• Machine 2 hosts Intelligence Server 1

• Machine 3 hosts Intelligence Server 2

When performing the installation, enter machine1.domain.com;machine2.domain.com;machine3.domain.com; on all three machines.

Example: Your environment contains the following five machines and you select three machines to host the communication agents:

• Machine 1 hosts MicroStrategy Library

• Machine 2 hosts Intelligence Server 1 and a communication agent

• Machine 3 hosts Intelligence Server 2 and a communication agent

• Machine 4 hosts Intelligence Server 3 and a communication agent

• Machine 5 hosts Intelligence Server or other services
Enter

machine1.domain.com;machine2.domain.com;machine3.domain.com;

on all machines.

Some important information to keep in mind:

- In MicroStrategy 2019, OpenSSH is no longer installed on Windows machines. You must install OpenSSH to start and stop services using topology. Upgrading your system from 11.0 to 2019 removes OpenSSH.

If consul is already installed on a machine, prior to installing MicroStrategy products, the service registration auto configuration and execution is affected. We do not recommend a separate consul installation on the same machine.

- If you select **Multiple machine environment**, you must enter an odd number of machines.

- If your environment includes more than one machine and machines with dynamic IP addresses as communication agents, use FQDN for the communication agent machine list during the installation.

- If your environment includes more than one machine and machines with more than one IP address as communication agents, it is recommended that you use FQDN for the communication agent machine list during the installation. If IP address list is used, make sure the machine with multiple IP addresses can be pinged with an IP address included in the list.

  Topology supports installation on machines with public IP addresses. To avoid unexpected joins to topology nodes, possibly exposing service information, you must configure your firewall correctly. Your firewall should allow ports for Server RPC (default 8300), and Serf LAN (default 8301), and Serf WAN (default 8302) only for the nodes within the Workstation topology node cluster. See *Enabling*
Deploying MicroStrategy Library through a WAR File

If you do not want to deploy MicroStrategy Library through the installer, which uses Tomcat by default, you can deploy MicroStrategy Library to a Web container.

To deploy Library and configure the registration service:

1. Deploy MicroStrategy Library to the desired web servers.

2. Once Library is deployed, the service registration agent that runs in this deployment becomes a monitoring agent by default. It automatically connects to the agent on the iServer machine that is connected to Library.

3. Make sure the Intelligence server, connected to Library, hosts the communication agent.

4. Keep in mind that restarting the application also restarts the service registration agent.

5. If the firewall is turned on for this machine, refer to Enabling Topology Communication Through a Firewall to make sure the proper ports are open to support service registration agent communications.

To create redundancy for the communication agents:

In this configuration, one machine hosts MicroStrategy Library and the other two machines host Intelligent servers (and/or more machines
hosting services). Redundancy is recommended, with a total of three communication agents.

1. Create a list of the three machines separated by semicolons.

librarymachine.domain.com;intelligenceserver1.domain.com;intelligenceserver2.domain.com

2. During the installation, enter the list of three machines, as noted above, on both Intelligent servers and/or any other machines where the MicroStrategy installation runs.

3. After deploying Library, manually change its configuration to run as a communication agent.
   a. Navigate to the deployed folder of MicroStrategy Library and the subfolder, \ServicesRegistration\yaml.
   b. Modify the configuration file.
   c. Restart all consul nodes.

Enabling Topology Communication Through a Firewall

MicroStrategy supports machines with registration services installed in the same subset as a MicroStrategy environment. If there is a firewall between MicroStrategy Services, make sure the specified ports (shown below) are allowed to send and receive TCP/UDP requests through the firewall, so the Services Registration agents can communicate properly. Consul uses both TCP and UDP, so make sure to allow both protocols.

- Server RPC (Default 8300)
  This port is used by the communication agents to handle incoming
requests from other agents. TCP only.

- **Serf LAN (Default 8301)**
  This port is used by all monitoring and communication agents to communicate with each other on the LAN. TCP and UDP.

- **Serf WAN (Default 8302).** Use this port for machine communication over a WAN. This port is used by servers to gossip over the WAN to other servers. TCP and UDP.

If you are performing a Windows installation and the firewall is on, the installer adds rules to open 8300 and 8301 for TCP and 8301 for UDP.

On machines with public IP addresses exposed to the internet, make sure firewall ports for Server RPC (default 8300), Serf LAN (default 8301), and Serf WAN (default 8302) are only for nodes within the Workstation topology cluster.

See the [Consul documentation](#) for more information on port requirements.

See [Using Firewalls](#) for more information about using firewalls with MicroStrategy.

**Removing a Service from the Topology Monitor**

The administrator may choose to install MicroStrategy products on a machine, but not plan to configure them at the moment. To eliminate confusion, the administrator can remove them from the topology monitor.

1. Navigate to /MicroStrategy/Services Registration/config in your installation directory.

2. Delete the .json file that corresponds with the name of the service you want to remove.
3. Navigate to the installation directory and delete the Services Registration/data folder.

4. Restart the consul.

5. In Workstation, navigate to the Topology view and click the Refresh button.

6. Verify that the services you removed are no longer visible.

**Restarting the Consul**

1. If you are using Windows, use the Service Manager to start and stop MicroStrategy Services Registration.

2. If you are using Linux, enter the following commands to start and stop the consul:

```
-> cd /Installationpath/MicroStrategy/install/ServicesRegistration/jar
-> java -jar svcsreg-admin.jar control consul start
```
Specifying the Topology Communication Agent

After installation, you can see all configuration files for the service's registration in the consul.json file, located in MicroStrategy/Services Registration/config, in your installation directory.

If you have a single machine environment or a multiple machine environment with two machines, consul.json appears as follows:

```json
{
    "enable_debug": true,
    "datacenter": "dc1",
    "enable_script_checks": true,
    "check_update_interval": "0s",
    "log_level": "WARN",
    "data_dir": "../data",
    "watches": [
        {
            "type": "key",
            "key": "envInfoTime",
            "handler_type": "script",
            "args": [
                "C:\\Program Files (x86) Common Files MicroStrategy JRE\\180_77\\Win64\\bin\\java",
                "-jar",
                "../jar/envinfo-collector.jar",
                "collect"
            ]
        }
    ],
    "server": true,
    "bootstrap_except": 1
}
```
If you have a multiple machine environment with three or more machines, consul.json appears as follows:

```json
{
    "enable_debug": true,
    "datacenter": "dc1",
    "enable_script_checks": true,
    "check_update_interval": "0s",
    "log_level": "WARN",
    "data_dir": ".../data",
    "watches": [
      {
        "type": "key",
        "key": "envInfoTime",
        "handler_type": "script",
        "args": [
          "C:\Program Files (x86)\Common Files\MicroStrategy\JRE\180_77\Win64\bin\java",
          "-jar",
          "../jar/envinfo-collector.jar",
          "collect"
        ]
      }
    ],
    "retry_join": [
      "was-jinli.corp.microstrategy.com",
      "wezhong-win1019.corp.microstrategy.com",
      "was-pcheng.corp.microstrategy.com"
    ],
    "server": true,
    "bootstrap_except": 3
}
```

In consul.json file, bootstrap_except, refers to the number of machines that act as communication agents in the environment. This is set on each machine where you specify the communication agent only. Specify the machines that act as communication agents in retry_join. This field is specified on every machine where the
MicroStrategy Services are installed so that they can find the communication agents in the network.

To support machines with dynamic and multiple IP addresses, the consul.json file is regenerated each time the service registration agent restarts.

Changing the Monitoring Agent to a Communication Agent

You can change a monitoring agent to a communication agent by manually modifying the installation_list.yaml file.

Manually modifying the installation_list.yaml file to change the agent from a monitoring to a communication agent (or vice versa) is not recommended. Only a MicroStrategy administrator should manually modify installation_list.yaml to correct input mistakes during installation or reconfigure the service registration agents in a cluster with manually deployed web applications.

During the installation, when selecting a multiple machine environment, the user must provide a list of communication agents. If the current machine where installation is running is not in the list, it is not configured as a communication agent. Instead, it is set as a monitoring agent only. If the library server is manually deployed on a machine, the embedded service registration agent is configured as a monitoring agent by default.

To manually change the monitoring agent to a communication agent:

1. After installation, navigate to the directory MicroStrategy/Services Registration/yaml.
2. Open installation_list.yaml and locate the consul section.

```yaml
---
consul:
install_path: "C:\\Program Files (x86)\\MicroStrategy\\Services Registration"
java_install_path: "C:\\Program Files (x86)\\Common Files\\MicroStrategy\\JRE\\180_192\\Win64\\bin"
retry_join:
- Wei2016.labs.microstrategy.com
- pcheng-W10-PRO.labs.microstrategy.com
- pcheng-RHEL-6-8.labs.microstrategy.com
server: false
---
```

3. Change the `server` value to `true`.

4. Add "bootstrap_except": 3 to the end of the file. If this is a two machine environment and both machines are started as monitoring agents, add "bootstrap_except":1 instead.

5. Make sure the machine name is added in the `retry_join` list and only three machine names are in the list. If this is a two machine environment, make sure only one machine name is in the list. Save the file.

6. Modify installation_list.yaml on all machines in the same consul cluster by including the name of this machine in the `retry_join` list and make sure only three machines are in the list. The list should be identical for all machines in the same consul cluster.

7. Stop the consul on all machines in the same consul cluster. These machines have the same `retry_join` list running as either a communication or monitoring agent.

8. Navigate to the installation directory and delete MicroStrategy/Services Registration/data.

9. Restart the consuls on all machines in the same consul cluster.
To switch one machine from a monitoring agent to a communication agent, another machine previously running as a communication agent in the same cluster should be switched to a monitoring agent. Do this by modifying the installation_list.yaml file. Change the server value to false and remove "bootstrap_except".

Starting and Stopping Services

The topology view uses SSH authentication to start and stop services remotely. If you are using Windows, the default Windows SSH configuration, TCP port 22, is used. The credentials are typically the machine administrator's credentials. See Configuring SSH on Windows to configure SSH on Windows. See Configuring SSH on Linux to configure SSH on Linux.

Configuring SSH on Windows

If SSH is not running or installed during your MicroStrategy installation, the installer does not enable the feature to start and stop services in Workstation. You must enable SSH server in Services Registration to start and stop services in Workstation.

To configure SSH on Windows:

1. Install or enable SSH server on your Windows machine. If your Windows machine has Windows 10 Fall Creators Update, Windows Service 1809, or Windows Server 2019, you can follow the steps from the following Microsoft websites:
   
   Open SSH in Windows 10!
   
   Windows Server 2019 Includes OpenSSH
   
   Using the OpenSSH Beta in Windows 10 Fall Creators Update and Windows Server 1709
2. To enable the built-in SSH server for older Windows versions, see Install Win32 OpenSSH.

3. Configure and start your SSH server.

4. If you have firewall, open the SSH port in your firewall. For example, port 22.

5. Create a SSH Server.json file using the example below and make the following modifications:

   - Modify "Port" to use your SSH server's port number.
   - Modify "consulInstallPath" to use your Services Registration installation folder.
   - Modify "version" to use your MicroStrategy version number.

```
{
    "service": {
        "Name": "SSH Server",
        "ID": "SSH Server",
        "Port": 22,
        "Check": {
            "ID": "MicroStrategy SSH Server Health Check",
            "Name": "MicroStrategy SSH Server Health Check",
            "Interval": "60s",
            "timeout": "5s",
            "args": [
                "cmd"
            ],
        }
    },
    "Tags": [
        "\"consulInstallPath\": "C:\\Program Files (x86)\\MicroStrategy\\Services Registration\",
        "\"version\": "11.1.0000.0076"
    ]
}
```
6. Copy the SSH Server.json file to the Services Registration installation folder. For example, 
/MicroStrategy\ServicesRegistration/config.

7. Restart MicroStrategy Services Registration from Windows Service Manager.

8. Open the Topology view in Workstation. Now you can start and stop services.

Configuring SSH on Linux

MicroStrategy uses the SSH included with your Linux OS by default. If SSH is not running at the time of your MicroStrategy installation, the installer does not enable the feature to start and stop services in the Workstation Topology view.

To configure SSH on Linux:

1. Install OpenSSH by opening a terminal and running the following commands with super user permissions.

   On Ubuntu/Debian/Linux Mint:
   
   ```
   # apt-get install openssh-server openssh-client
   ```

   On RHEL/Centos/Fedora:
   
   ```
   # yum -y install openssh-server openssh-clients
   ```

2. Start the service by typing the following commands in terminal:

   ```
   # chkconfig sshd on
   # service sshd start
   ```

3. If you have firewall, open the SSH port in your firewall. For example, port 22.
4. Create a SSH Server.json file using the example below and make the following modifications:

- Modify "Port" to use your SSH server's port number.
- Modify "consulInstallPath" to use your Services Registration installation folder.
- Modify "version" to use your MicroStrategy version number.

```json
{
  "service": {
    "Name": "SSH Server",
    "ID": "SSH Server",
    "Port": "s",
    "Check": {
      "ID": "MicroStrategy SSH Server Health Check",
      "Name": "MicroStrategy SSH Server Health Check",
      "Interval": "60s",
      "timeout": "5s",
      "args": ["/bin/sh"]
    },
    "Tags": [
      "consulInstallPath": "/opt/MicroStrategy/ServicesRegistration",
      "version": "11.1.0000.0076"
    ]
  }
}
```

5. Copy the SSH Server.json file to the Services Registration installation folder. For example, `/opt/MicroStrategy/ServicesRegistration/config`.

6. Restart MicroStrategy Services Registration using the following command:

```
# java -jar <MSTR_INSTALL_PATH>/ServicesRegistration/svcsreg-admin.jar control consul restart
```
7. Open the Topology view in Workstation. Now you can start and stop services.

## Supported MicroStrategy Platform Services

<table>
<thead>
<tr>
<th>Old Service Display Name</th>
<th>New Service Display Name</th>
<th>Service Description</th>
<th>Default Port</th>
<th>Executable</th>
<th>Workstation Start/Stop Supported</th>
<th>Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligence Server</td>
<td>Intelligence</td>
<td>Executes the query, reporting and analysis server for MicroStrategy software. Controls security, caching, clustering, queueing and prioritization.</td>
<td>34952</td>
<td>MSTRSv2_64.exe</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>PDF Export Engine</td>
<td>Export</td>
<td>Used by Intelligence to create PDF files</td>
<td>20100</td>
<td>MSTRPDFExportService.exe</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| Library Server           | Library                  | A personalized portal for every end user to access all their dossiers and report services documents (documents). BI authors can easily share links to dossiers | Tomcat: 8080  
https:8443 | MicroStrategyLibrary.war | No                | No         |
<table>
<thead>
<tr>
<th>Old Service Display Name</th>
<th>New Service Display Name</th>
<th>Service Description</th>
<th>Default Port</th>
<th>Executable</th>
<th>Workstation Start/Stop Supported</th>
<th>Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>or documents, which are hosted in MicroStrategy Library. End users can log into MicroStrategy Library to find a list of all dossiers and documents that they have access to.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collaboration Server</td>
<td>Collaboration</td>
<td>Allows teams, departments, and organizations to</td>
<td>3000</td>
<td>server.js</td>
<td>Yes</td>
<td>Collaboration Store</td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>boost analytics adoption via interactive discussion threads and an intuitive comments panel. With real-time alerts, notifications, and easily shared personalized views, users have a holistic view of their business and benefit from</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>-------------------</td>
<td>--------------</td>
<td>------------</td>
<td>-----------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Mobile Server</td>
<td>Mobile</td>
<td>An interactive interface of the MicroStrategy BI platform that lets mobile business users harness the analytical power of MicroStrategy through the use of their iPhone®, iPad®, and Android™</td>
<td>http: 80</td>
<td>Admin.aspx</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Collaboration Repository</td>
<td>Collaboration Store</td>
<td></td>
<td>27017</td>
<td>mongod.exe</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Suppored</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Web Server</td>
<td>Web</td>
<td>A browser-based client interface that enables IT to satisfy the diverse analytics needs of every enterprise user—consumers, analysts, data scientists, developers, and administrators</td>
<td>Tomcat: 8080, https: 8443</td>
<td>MicroStrategyMobile.war</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Web Server</td>
<td>Web</td>
<td>http: 80, https: 443</td>
<td>main.aspx</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Messing Server</td>
<td>Telmetry</td>
<td>ators—without the burden of distributing and maintaining multiple software packages. A distributed streaming platform that stores records streams of data in topics. Kafka producers allow an application, such as Identity and Intelligen</td>
<td>8080</td>
<td>MicroStrategy.war</td>
<td>No</td>
<td>Telemetry Manager</td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>---------------------</td>
<td>---------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Messaging Manager</td>
<td>Telemetry Manager</td>
<td>The MicroStrategy platform messaging service that coordinates the Kafka Cluster</td>
<td>2181</td>
<td>zookeeper.exe</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Usher Database Producer</td>
<td>Identity Telemetry</td>
<td>Sends the data generated by Identity</td>
<td>N/A</td>
<td>UsherMetadataProducer.exe</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------------------</td>
<td>----------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Platform Analytics Consumer</td>
<td>Telemetry Store</td>
<td>Reads the data that Intelligen ce and Identity send to Telemetry, transforms the data, and loads it into the Platform Analytics warehouse.</td>
<td>N/A</td>
<td>UsherAnalyticsConsumer.exe</td>
<td>Yes</td>
<td>Telemetry Server</td>
</tr>
<tr>
<td>In-memory Cache</td>
<td>Telemetry Cache</td>
<td>An in-memory cache used to optimize performance for the</td>
<td>6379</td>
<td>redis-server.exe</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Repository</td>
<td>Platform Analytics Consumer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Server</td>
<td>Identity</td>
<td>A highly scalable, flexible, and easy to maintain system built on industry standards such as Security Assertion Markup Language (SAML), Open Authorization (OAuth) 2.0 and OpenID</td>
<td>ssl: 1443 two-way ssl: 2443</td>
<td>mysqld.exe</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>----------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Connect. The server synchronizes identities with enterprise identity management (IDM) systems of record, and presents those identities to Badge clients for authentication. The Identity server can be installed on premises or in the MicroStrat...</td>
<td>Workstation Start/Stop Supported</td>
<td>Executable</td>
<td>Dependency</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>----------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Gateway Server</td>
<td>Identity Gateway</td>
<td>A mechanism (also known as a connector) that permits communication between Identity server and a secured asset. It represents a bi-directional secure communication connection used by the Identity server to communicate.</td>
<td>two-way ssl: 9501</td>
<td>gateway.war</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each gateway is deployed as a web service which ensures secure communication between the Identity server, IDM systems, and PAC systems, and

cate with user repositories hosted in IDM systems and PACS, and for logical resources.
<table>
<thead>
<tr>
<th>Old Service Display Name</th>
<th>New Service Display Name</th>
<th>Service Description</th>
<th>Default Port</th>
<th>Executable</th>
<th>Workstation Start/Stop Supported</th>
<th>Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Manager</td>
<td>Identity Manager</td>
<td>The group of users in your organization who can use the Badge app on their smartphone to validate their identity, log on to applications, gain access to secure physical resources, and so on. Identity Manager is the administrator</td>
<td>Windows: 443</td>
<td>networkmgr httpd.exe</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Certificate Manager</td>
<td>Certificate Store</td>
<td>Highlight services lacking SSL support, tracks</td>
<td>5050</td>
<td>CertificateManager.exe</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Old Service Display Name</td>
<td>New Service Display Name</td>
<td>Service Description</td>
<td>Default Port</td>
<td>Executable</td>
<td>Workstation Start/Stop Supported</td>
<td>Dependency</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>current and pending certificate expirations. Presents visual cues about expiring or already expired certificates across the entire MicroStrategy deployment.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Starting in version 2019, some of the service names have been changed to better explain their behavior. See the online readme corresponding to your software version for a list of the changes.

MicroStrategy Logging Consumer, MicroStrategy Listener, SSH, and MicroStrategy Registration Service are not monitored.
If using clusters, make sure that all machines in the Web server cluster can communicate with all machines in the Intelligence Server cluster.

Modification of the port the agents use to communicate is currently not supported.

Troubleshooting Topology

This section covers issues that may occur in Workstation and how to correct them.

How to update service ports for the service registration/topology monitor

1. Navigate to the service configuration folder at /MicroStrategy/ServicesRegistration/config. Edit the file for the service running with a non-default port number by updating the run-time port number being used.

```json
{
    "service": {
        "Name": "MySQL",
        "ID": "MySQL",
        "Port": 3306,
        "Tags": [
            {
                "consulInstallPath": "C:\Program Files(x86)\MicroStrategy\Services Registration",
                "typeId": 13,
                "version": "11.1.0000.0112",
                "start": "net start MySQL /y",
                "stop": "net stop MySQL /y"
            }
        ],
        "Check": {
            "ID": "MySQL Health Check",
            "Name": "MySQL Health Check",
            "tcp": "localhost:3306",
            "Interval": "10s",
            "timeout": "5s"
        }
    }
}
```
2. Save the file and restart the service registration agent.

Service registration does not work when there is no network connection

If there is no network connection after installation, the service registration still works for all local services. If not, make sure the user logging into the machine is a local administrator. A user with adequate privileges for service registration, when connected to the network, might not have full administrator privileges locally when there is no network connection.

The service registration agent cannot start and *No private IPV4 address found* is found in the log file

1. If the machine has more than one IP address, see *Configuring Environments with Multiple IP Addresses*

2. If the machine has only one IP address, check the network setting using the following command, if the IPv4 is *not* in the list of acceptable private (RFC1918) addresses provided in the table below.

   On Windows:
   
   ```
   ipconfig/all
   ```

   On Linux:
   
   ```
   ifconfig -a
   ```
3. Check whether the network configuration is intended. If so and you want to use service registration, complete steps 4 and 5 for a workaround to bind the service registration agent to the non-acceptable IPv4 address.

4. Navigate to the Services Registration\yaml folder.

5. Edit the installation_list.yaml file by adding `bind_addr: x.x.x.x` to the consul section. Make sure to include a space between the colon and the IP address.

```
consul:
  install_path: "C:\\Program Files (x86)\\MicroStrategy\\Services Registration"
  java_install_path: "C:\\Program Files (x86)\\Common Files\\MicroStrategy\\JRE\\180_192\\Win64\\bin"
  server: false
```
retry_join:
- sap-win2016.labs.microstrategy.com
- pcheng-W10-PRO.labs.microstrategy.com
- pcheng-RHEL-6-8.labs.microstrategy.com
bind_addr: 10.21.20.222

6. Stop the service registration agent, delete the Services Registration\data folder, and restart the service registration agent.

How to check the service registration cluster status

1. Open a command window and go to your MicroStrategy install path at \Service Registration\bin.

2. Execute the following command.

   On Windows:
   ```
   consul members
   ```

   On Linux:
   ```
   ./consul members
   ```

3. Verify the returned list to see whether it is consistent with the cluster configuration. Look at machine names and number of machines with communication agents running, machine names and number of machines with monitor agents running, status of these agents. etc.

"MicroStrategy Services Registration is NOT available, please check the service status."

While using Workstation Topology to monitor services, an empty page appears with the message, "MicroStrategy Services Registration is
NOT available, please check the service status." Follow the steps below to troubleshoot this issue.

Is the service registration agent running on the Library machine to which Workstation is connecting?

Log into the machine where Library is installed and check whether MicroStrategy Services Registration is running. On Windows, use the Windows Service Manager. On Linux, use `ps -ef | grep consul`.

If the agent is running, check whether the cluster agent is forming successfully. See How to update service ports.

1. If there is a firewall on the machine in which Library is installed, make sure the 8300(TCP) and 8301(TCP/UDP) ports are open.

2. Try to ping the machines listed in Services Registration/config/consul.json for "retry_join" to ensure the machines can be accessed.

3. Make sure the leader election is complete. Check the ServicesRegistration log file with the latest time stamp in the Services Registration/log folder. If you see the message, "failed to sync remote state: No cluster leader" follow steps 4 and 5. Otherwise, skip to step 6.

4. The correct cluster configuration should be verified on every machine with a service registration agent. Check the configuration of the cluster. There should be one or three communication agents defined in the cluster as specified in the "retry_join" list of the installation_list.yaml file. This list should be consistent on all nodes.

5. Make sure there is no dirty data in the Services Registration/data folder. After unsuccessful cluster forming or modification of cluster members, there may be invalid data left in the data files.
Stop all agents, delete the Services Registration/data folder on every cluster node and restart all agents to form the cluster.

6. If MicroStrategy Library is deployed on a machine where MicroStrategy Services are not installed, go to the Application Server managing the consul and restart the MicroStrategy Library deployment. If the agent is not running, check the ServicesRegistration log file with the latest time stamp in the Services Registration/log folder to see why the agent did not start. If the message, "No private IPv4 address found" appears, see The service registration agent cannot start.

If the agent is not running, check the ServicesRegistration log file with the latest time stamp in the Services Registration/log folder to see why the agent cannot start.

Does the machine have multiple IP addresses? Were all IPs available when installing the product?

Does the machine have an unacceptable IP address? If the agent hit "No private IPv4 address found," see The service registration agent cannot start.

How was the MicroStrategy Library WAR file deployed on the server?

There may be issues if the Linux server was not upgraded properly. For example, if you upgraded to 11.0 from 10.10 on a Linux server, did not run the MicroStrategy installer, but downloaded and copied the WAR file. In this situation, see Install MicroStrategy in the In-Place Upgrade on a Linux Deployment section of the Upgrade Guide and Deployment Scenarios in the Installing MicroStrategy Library on Linux section of the Upgrade Guide.

Are the nodes communicating with each other through network address translation (NAT)?

NAT is currently not supported by the service registration feature.
A service displays a red icon

1. In Workstation, hover over the stop icon to view additional information in the tooltip.

2. Review the possible causes provided in the tooltip.

3. Verify the service is running. Log into the machine where the service is installed. Verify it is running in Windows Service Manager.

4. Verify the required port is available. If the service fails to start, check to see whether the required port is available. See Supported MicroStrategy Platform Services to view the default ports.

5. Verify the proper port is in the configuration file. If the service is running, but Workstation shows the service is down, check the configuration file (such as MicroStrategy Mobile Server JSP.json) for this service. The configuration file is located in the installation directory at MicroStrategy/Services Registration/config. Check to see if the port number, http port number, and https port number are consistent with the port used for the service.

6. If the ports are correctly defined in the service.json file, but Workstation still shows the service is down, copy the URL for the health check from the service.json file. Paste the URL in your browser to see if it is a valid link. If the link does not work, replace "localhost" with the IP address of the machine from which you are accessing the URL and try again.

7. Check the log file at /MicroStrategy/Services Registration/log in your installation directory.

If a service is configured to use a non-default port number after the installation, the new port number is not recorded in the JSON
configuration file. You must manually update the JSON file with the post-install configured port number and restart the MicroStrategy Service Registry. Please refer to How to Update Services Ports.

Unable to start or stop a service

1. Connect to an environment, making sure the corresponding user login has adequate privileges. In Workstation, log into an environment as a MicroStrategy user with administrator privileges or as an administrator user of the web container for the Library. In maintenance mode, go to the topology view and verify that start/stop capability is supported for a service, by right-clicking on the service. If the Start and Stop options in the pop-up menu are grayed out, the start/stop capability is not supported for that service.

2. Verify SSH is available and running on the machine with the service. If not, see Configuring SSH on Windows or Configuring SSH on Linux.

3. Verify the firewall is enabled and port 22 is open on the firewall.

4. Verify the service status in a refreshed view. If a user has sent the start or stop request in Workstation, before another request is sent out, refresh the topology view to get the latest service status.

A service is not reachable

1. Verify that the machine where the service is hosted is reachable from the MicroStrategy Library machine.

2. If there are firewalls in the environment, verify that the ports (8300 and 8301) are open to support sending and receiving TCP/UDP requests between the machines hosting the services.
3. Verify that the agent is running. On Windows, use the Windows Service Manager to check whether MicroStrategy Services Registration is running. On Linux, use `ps -ef | grep -i consul`.

Consul does not start after installing M2019 Update 2 Hotfix on a Linux machine

The `nslookup` tool is missing if Redhat Enterprise Linux 7 is installed using the minimal option. Consul depends on the `nslookup` command to resolve a machine’s IP address from the fully-qualified domain name (FQDN). To ensure that Consul can be started successfully, install the missing tool by executing the following command in a terminal:

```
# yum install bind-utils
```

Customized Deployments

**Case 1:** When installing topology in a two machine environment, MicroStrategy services are installed without the multiple machines option. As a result, you will need to manually configure the consul cluster after the installation. In this case, Library was deployed through the installer and both the Library and Web servers are on the same machine. iServer is installed on another machine. First, make sure the required ports are open.

On the Library server machine:

1. Navigate to the MicroStrategy/Services Registration/yaml directory.
2. Open `installation_list.yaml` and locate the consul section.
3. Verify the `server` parameter is set to `true`.
4. On Windows, use the Service Manager (Run > services.msc) to open services. Verify that MicroStrategy Services Registration is running.

On the Intelligence server machine:
1. Navigate to the MicroStrategy/Services Registration/yaml directory.
2. Open installation_list.yaml and locate the consul section.
3. Add the retry_join parameter with the FQDN of the Library server, so the Intelligence server can point to it.
4. Make sure the format for retry_join matches that shown in Changing the Monitoring Agent to a Communication Agent.
5. Remove the entire bootstrap_except line, including the parameter name, and change server: true to server: false.
6. Save the file.

On both machines:
1. Stop the consul on all machines. In this case, there are two machines.
2. On Windows, use Service Manager (Run > services.msc) to stop MicroStrategy Services Registration.
3. On Linux, refer to Restarting the Consul to stop MicroStrategy Services Registration.
4. Navigate the installation directory and delete MicroStrategy/Services Registration/data.
5. Restart the consuls on all machines in the same consul cluster.
Case 2: JBoss is running on the same machine as the Intelligence server. Both Web and Library are installed, but deployed in JBoss.

1. Since Web and Library are on the same machine, both MicroStrategy Web Server JSP.json and MicroStrategy Library REST Server.json are available in the ServicesRegistration/config install folder. Update both files with the port number used for the Web and Library JBoss deployment, if necessary.


3. Stop the consul and delete the ServicesRegistration/data folder.

4. Restart the consul.

Configuring Environments with Dynamic or Multiple IP Addresses

Automatic configuration of agent clustering is limited for machines with dynamic or multiple IP addresses. In some cases, manual configuration is required. See Configuring Environments with Nodes Using Dynamic IP Addresses and Configuring Environments with Multiple IP Addresses for more information.

Configuring Environments with Nodes Using Dynamic IP Addresses

If you are using a single machine environment with a dynamic IP address and connecting Workstation to Library, you may encounter
errors when accessing the Topology screen in Workstation.

After installation is finished and the machine is restarted, the communication agent may start before the network IP is fully resolved and ready on the machine. Therefore, the communication agent may obtain a temporary IP during the startup and consider it as the fully resolved one. Using the wrong IP on the machine affects the communication agent’s function. If this problem continues to occur, a manual restart of the consul is required. See Restarting the Consul for more information.

We do not recommend using machines with dynamic IP addresses as communication agents in a cluster. If your environment includes more than one machine and machines with dynamic IP addresses are used as communication agents, use FQDN (such as machine1.domain.com) in the communication agent machine list during installation.

If FQDN is not used, you should manually update the "retry_join" section in consul.json on all machines with communication and monitoring agents when all machines with dynamic IP addresses are fully resolved, every time those machines get new IPs.

```json
{
    "enable_debug": true,
    "datacenter": "dc1",
    "enable_script_checks": true,
    "check_update_interval": "0s",
    "log_level": "WARN",
    "data_dir": ".\data",
    "watches": [
        {
            "type": "key",
            "key": "envInfoTime",
            "handler_type": "script",
            "args": [
                "C:\Program Files (x86)\Common Files\MicroStrategy\JRE\180_77\Win64\bin\java",
                "-jar",
            ]
        }
    ]
}
```
Configuring Environments with Multiple IP Addresses

We do not recommend using machines with more than one IP address as a communication agent. If your environment includes one or more machines and machines with more than one IP address are used as communication agents, make sure all IPs are available when installing. The IP mapping to the FQDN of the machine is included in the consul.json file, as shown below.

```
{
    "enable_debug": true,
    "datacenter": "dc1",
    "enable_script_checks": true,
    "check_update_interval": "0s",
    "log_level": "WARN",
    "data_dir": ".\data",
    "watches": [  
    {  
        "type": "key",
        "key": "envInfoTime",
        "handler_type": "script",
        "args": [  
            "C:\\Program Files (x86)\\Common Files\\MicroStrategy\\JRE\\180_77\\Win64\\bin\\java",
            "-jar",
            ".\jar\envinfo-collector.jar",
            "collect"
        ]
    },
    ]
    },
    "retry_join": [
        "10.27.16.239",
        "localhost",
        "10.27.17.238"
    ],
    "server": true,
    "bootstrap_except": 3
}
```
If a machine has two available IP addresses, the consul may bind itself to one IP mapping for FQDN, while other machines use FQDN mapping to another IP. In this situation, you should manually modify consul.json. The IP used for "bind_addr" should be consistent with the IP the other machine finds using the FQDN; the IP for "bind_addr" and the IP (if not using FQDN) in "retry_join" list representing this machine should be consistent. After all manual modifications of consul.json are complete on all machines with consul installed, stop the consul on all machines, delete the data folder in path/MicroStrategy/Services Registration on every machine, and restart all consuls.

If MicroStrategy is installed when only one IP is available and runs later when multiple IPs are available, the consul will not start. The ServiceRegistration.log displays the error, "Multiple private IPv4 addresses found. Please configure one."

Using FQDN for the communication agent machine list during the installation is recommended. If the IP address list is used during installation, make sure to use the IP mapping to the FQDN for the machine with multiple IPs.
Machines with more than one IP address, as well as dynamic IPs, are not supported for automatic clustering.

Certified Operating Systems for Topology

The following operating systems are certified for topology:

- Windows Server 2008 R2 SP1
- Windows Server 2012 R2
- Windows Server 2016
- Windows 7
- Windows 10 October Update 2018
- Red Hat Enterprise Linux 6.x
- Red Hat Enterprise Linux 7.x (including SE Linux)
- SUSE Linux 12.x
- SUSE Linux 15.x
- Amazon Linux 2

Certified Web Containers for Topology

The following web containers are certified for topology:

- Apache Tomcat 8.5.30
- Apache Tomcat 9.0.12
- WebLogic 12cR2 (12.2.1.3.0)
- WebSphere Application Server 9.0
- Wildfly 13
• Wildfly 14
• JBoss EAP 7.1
• Jetty 9.4

See *How to Set Up Application Servers* for more information about setting up application servers.
Installing and Configuring the Identity Messaging Service

Identity Messaging is included with the MicroStrategy platform when the MicroStrategy Badge component is selected for installation. The Identity Messaging service acts as a scalable messaging gateway between the Identity server and email, SMS, and push notification providers.

The Identity Messaging Service is currently certified for MicroStrategy 2019 Linux deployments.

Identity Messaging Service Dependencies

The Identity Messaging service makes use of the following 3rd party components. Those components will need to be installed prior to installing the Identity Messaging service:

<table>
<thead>
<tr>
<th>3rd Party Component</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Tomcat</td>
<td>8.0 or later</td>
<td>Apache Tomcat 8.5 is recommended</td>
</tr>
<tr>
<td>Oracle Java Runtime Environment (JRE)</td>
<td>1.8 or later</td>
<td>IBM JDK is not supported</td>
</tr>
</tbody>
</table>
### 3rd Party Component

<table>
<thead>
<tr>
<th><strong>3rd Party Component</strong></th>
<th><strong>Version</strong></th>
<th><strong>Notes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>MongoDB</td>
<td>3.6.2 or later</td>
<td>MongoDB 3.6.5 is recommended</td>
</tr>
</tbody>
</table>

## Installing the Identity Messaging Service on Linux

The Identity Messaging service components will be installed with your installation of MicroStrategy Badge.

After completing the MicroStrategy 2019 installer process, the MicroStrategy Identity Messaging service WAR file can be found in:

<installation path>/MicroStrategy/install/mercury.war

### Preparing the Push Notification Credentials Folder

To be able to send push notifications to mobile devices, the Identity Messaging service needs access to the Apple and Google push notification credentials.

1. Create a the following folder structure under the /opt/usher/usherserver/db_configs/identitymessaging/ directory:
   
   /apns
   
   /gcm
   
   /fcm

2. Place the two Apple push notification files in the /apns folder.

3. Place the Google GCM file in the /gcm folder.

4. Place the Google FCM file in the /fcm folder.
Preparing the Service Folder Structure

The MicroStrategy Identity Messaging micro service leverages Apache Tomcat as the service container.

New Installations

The following folder structure will be created during installation:

/bin
/conf
/logs
/run
/temp
/webapps
/work

Existing MicroStrategy Identity Deployments

To set up the right folder structure from your existing Linux-based MicroStrategy Identity Installation (assumed to be in /opt/usher):

1. Duplicate the existing MicroStrategy Identity Server application folder:

```bash
cd /opt/usher/usherserver/usherApps/
cp -pr shardIDM shardIdentityMessaging
cd shardIdentityMessaging
rm -rf webapps/*
rm -f logs/*
```
2. Modify the `conf/server.xml` file:

- Change remote shutdown port to a port not in use. Make sure there is no port conflict. For example:

  ```xml
  <Server port="7501" shutdown="SHUTDOWN">
  ```

- Change the connector to listen on a different HTTP port, e.g. 5443. Modify the one with `certificateVerification="none"`, delete the other connector. e.g. `<Connector port="5443" protocol="org.apache.coyote.http11.Http11NioProtocol"`

3. Copy the `mercury.war` file to the `/webapps` folder.

### Configuring the Identity Messaging Service

The configuration settings described below should be made to the `application-prod.yml` file. This file is located in the `/opt/user/` There is no need to modify the `application.yml` file found in the same location.

### MongoDB Configuration

```yaml
spring:
data:
  mongodb:
    # uri: mongodb://mstr:<password>@127.0.0.1:27017/mercuryDb?authSource=admin
    host:127.0.0.1
    port:27017
    username: mstr
    password: <password>
    database: mercuryDb
```

### Update JWT Secret

```yaml
jhipster:
  security:
```
For security measures we suggest use the following script to generate the secret:

```
#!/bin/bash
token_size=66
openssl rand -base64 ${token_size} | tr -d '
'; echo;
```

Copy the output, not including "(stdin)=" if it exists, and paste it in the secret section in application-prod.yml as shown below. If you have one than more mercury nodes, please copy the SAME value to other nodes.

### Third-party Service Credential Registration

For third-party services to access and leverage the service, credentials need to be registered. Append the credentials to the application-prod.yml file in the application/credential section with the following format:

```
application:
  credential:
    <username>: <userpassword>
```

The username and user password fields are case sensitive. If your password includes any of the characters ":-{}[][]!#$>@", make sure you enclose the password string in single or double quotes. If you need to use single or double quotes in the password, escape them as well (a single quote in a single-quote-enclosed string is escaped by doubling it, a double quote in a double-quote-enclosed string is escaped by being prefixed by a backslash). You should also quote your string if the password can be mistaken for a number or a literal boolean (true, false, yes, no, etc.).
Email Gateway Configuration

For the service to start, the email gateway needs to be setup. Include the following parameters based on your configuration:

```
simplejavamail:
  transportstrategy: SMTP
  opportunistic: #Optional
    tls: true
  smtp:
    host: <ip or dns name for SMTP server>
    port: <SMTP Server port number; usually 25>
    username: <username for SMTP server is an authentication is required>
    password: <user password for SMTP server is an authentication is required>
    # proxy: # Optional: Sock 5 proxy configuration
    # host: <Sock5 proxy server ip or name>
    # port: <Sock5 proxy server port
    # username: <Optional: Sock5 username name>
    # password: <Optional: Sock5 user password>
    # socks5bridge: # Optional: This section is only relevant if
    # port: 1081 # an unused port needs to be specified. If not
```

Add email whitelist for the messaging service:

```
application:
  ...
  emailWhiteList:
    - <email1>
    - <email2>
```

Mobile App Push Notification Gateway Configuration

In the `application-prod.yml`, include the following entries:

```
application:
  apns:
  keypath: <path to APNS certificate file>
  password: <path to text file that contains the password for the
  fcmkeypath: <path to FCM token file>
  gcmkeypath: <path to GCM token file>
```
SMS Gateway Configuration

SMS configuration can be inserted on-the-fly after Identity Messaging service is running. See *Identity Server Configuration to Work With Identity Messaging Service* for details.

Proxy Configuration

If requests are routed through a proxy, both the application proxy setting and JVM proxy setting need to be configured. Application proxy setting is set by adding the following in `application-prod.yml`:

```
application:
  proxy:
    enabled: false
    host:
    port:
    username:
    password:
```

Modify Startup Script `tomcat.sh`

The `/bin` folder contains the service startup script for Linux.

1. **Edit the** `shardIdentityMessaging/bin/tomcat.sh` **file to replace both instances of** `shardIDM` **with** `shardIdentityMessaging`.

2. **Change the** `jmxremote port` **to a port other than 5500 to make sure there is no port conflict**.

3. **Make sure your startup script includes** –
   `Dspring.profiles.active=prod` **and** –
   `Dspring.config.location=file:<path_to_config_files>/application.yml,file:<path_to_config_files>/application-prod.yml`, **as parameters to either java_ops or catalina_opts environment variable.**
4. To setup JVM proxy, include 

```
-Dhttps.proxyHost=<ip> -Dhttps.proxyPort=8080
```

in CATALINA_OPTS

```
```

CATALINA_PID="/opt/usher/usherserver/usherApps/shardIdentityMessaging/run/tomcat.pid"

CATALINA_OPTS="${CATALINA_OPTS} -Dspring.config.location=file:/opt/usher/usherserver/db_configs/identitymessaging/application.yml,file:/opt/usher/usherserver/db_configs/identitymessaging/application-prod.yml"

CATALINA_OPTS="${CATALINA_OPTS} -Dspring.profiles.active=prod"

# JVM proxy setting

```
CATALINA_OPTS="${{CATALINA_OPTS} -Dhttps.proxyHost=<ip> -Dhttps.proxyPort=8080}
```

Managing the service

- Start the service:
  
  `/opt/usher/usherserver/usherApps/shardIdentityMessaging/tomcat.sh start`

- Stop the service:
  
  `/opt/usher/usherserver/usherApps/shardIdentityMessaging/tomcat.sh stop`

How to Monitor the Identity Messaging Service Deployment

The Identity Messaging Service can be monitored by making an API call and confirming that the output matches the expected one. In a multi-node deployment, it is recommended to monitor each node individually as well as the load balancer that sits in front of them.

The API is a REST API that can be called from monitoring tools or command line utilities like curl and wget.

`curl`
https://<hostname>:<hostport>/mercury/management/health

The expected response is
{"status":"UP"}

Troubleshooting the Identity Messaging Service Installation

The Tomcat log file catalina.out will contain most of the web application container level errors.

Identity Server Configuration to Work With Identity Messaging Service

Add URL and API Information of the Identity Messaging Service

Update the MicroStrategy Identity Server serverdb.conf, located at <usher installation directory>/usherserver/db_configs/usher/password/serverdb.conf, to include an entry for "mercury.urls" (replace the three placeholders with the values appropriate for your environment; refer to the "3rd party service credential registration" section above to setup or retrieve the user credentials)

```
```
Settings for Account based notification

The following settings should be included in serverdb.conf. In the account based installation, Identity Server requires them to work with Identity Messaging service.

```
email.fromAddr.default = "<the sender email addresses appears in email>"
email.sendOtp.subject.default = "<The subject of email contains an OTP code for a user to input in a verification process>"
email.sendOtpLink.subject.default = "<The subject of email contains a link for a user to click to verify their email address>"
email.iot.subject.default = "<The subject of email to invite user>"
mobile.sms.identifier = "<The identifier of SMS sender in Mercury which needs to be configured in Identity Messaging Service first. The following example is for Twilio setup:>
```
curl --request POST \
--url https://<identity_messaging_host>:<port>/mercury/api/user/<the identifier you want to use for mobile.sms.identifier>/smsconfig/twilio \
--header 'authorization: Bearer <token obtained from authenticate API>' \
--header 'content-type: application/json' \
--data '{"twilio_accountSid": "<twilio account sid>"}'
```

- The `mobile.sms.identifier` needs to be configured in Identity Messaging Service first. For example:

```
curl --request POST \
--url https://<identity_messaging_host>:<port>/mercury/api/templateString/<the identifier you want to use for mercury.template.logout.asNewLogin.iOS/Android>/en \
--header 'authorization: Bearer <token obtained from authenticate API>' \
--header 'content-type: application/json' \
--data '{"title":"Multiple devices detected","body":"Our records indicate that your MyMazda account is being accessed across multiple devices. You have been logged out due to security reasons."}'
```

- Authenticate API to obtain identity messaging service jwt token.
refer to *Third-party Service Credential Registration* to setup or retrieve the user credentials.

```
curl --request POST \
--url https://<identity_messaging_host>:<port>/mercury/api/authenticate \
-H 'Content-Type: application/json' \
-d '{
    "username": "<service name>",
    "password": "<service password>",
    "rememberMe": true
}'
```
This section describes processes that enable administrators to manage MicroStrategy Identity.

Managing MicroStrategy Identity Administrators

Using MicroStrategy Identity Manager you can add or remove Identity Admins. All admins have the ability to log into MicroStrategy Identity Manager to access the MicroStrategy Identity Configuration, create new MicroStrategy Identity Networks, and manage other Identity Admins.

How to add an Identity Administrator

1. Log into MicroStrategy Identity Manager:
   1. In a web browser, navigate to your organization's Identity Manager home page. The MicroStrategy Identity Manager page opens.
   2. Open the Scanner tab on your MicroStrategy Badge app.
   3. Scan the QR code displayed on the MicroStrategy Identity Manager page to log into your network.

2. Click the Administrator button in the upper-right. The Administrators page opens.

3. Click the + Add Administrator(s) button in the upper-right.

4. In the pop-up, enter the new administrator's email address and click Add.

5. In the next pop-up, enter the administrator's first and last name and click Add.

The new Identity Admin is added to the list of administrators and will receive instructions for acquiring the admin badge.
How to delete an Identity Administrator

1. Log into MicroStrategy Identity Manager:
   1. In a web browser, navigate to your organization's Identity Manager home page. The MicroStrategy Identity Manager page opens.
   2. Open the Scanner tab on your MicroStrategy Badge app.
   3. Scan the QR code displayed on the MicroStrategy Identity Manager page to log into your network.

2. Click the Administrator button in the upper-right.

3. Click the check box next to the admin(s) that you want to delete.

4. Click - Delete Administrator(s) in the upper-right.

5. In the pop-up, click Delete.

Managing the Signing Certificate Authority

The MicroStrategy Identity Server uses a self-signed SSL certificate as a Certificate Authority (CA) to sign client certificates used by other MicroStrategy Identity components. This includes the MicroStrategy Identity Agent, and custom applications implementing MicroStrategy Identity APIs.

- For Windows implementations, the MicroStrategy Installation Wizard automatically generates a Signing CA certificate that is valid for 1 year, and propagates it to the required system locations so that Tomcat will trust this CA.

- For Linux implementations, you must generate and propagate the Signing CA certificate manually.
Regardless of the host when the Signing CA certificate expires, or is about to expire, you must manually regenerate and propagate a new certificate.

You can create the Signing CA using OpenSSL®.

If you copy and paste the OpenSSL commands shown below, please ensure that the option flags are marked with a hyphen/minus sign instead of an en dash or em dash. If any hyphens are autoformatted into different characters, then they will be parsed incorrectly on the command line and you will get "unrecognized option" errors.

- For **Windows**, an OpenSSL utility is installed along with MicroStrategy Identity.

  The default file locations are as follows:

  - **Signing CA certificate**: C:\Program Files (x86)\Common Files\MicroStrategy\certificates\ushersigningca.crt
  - **Signing CA private key**: C:\Program Files (x86)\Common Files\MicroStrategy\keys\ushersigningca.key
  - **64-bit Java CA store**: C:\Program Files (x86)\Common Files\MicroStrategy\JRE\180_77\Win64\lib\security\cacerts
  - **OpenSSL executable**: C:\Program Files (x86)\Common Files\MicroStrategy\OpenSSL\openssl-1.0.2e\openssl.exe

- For **Linux**, an OpenSSL utility is included with many distributions.

---

**Generate a new Signing Certificate Authority**

Follow these steps if you do not already have a Signing CA private key and certificate (for example, if you are setting up MicroStrategy
Identity for the first time on Linux).

Use OpenSSL to generate a private key (.key) and certificate (.crt).

- For Windows:
  1. Run Command Prompt as administrator.
  2. Navigate to the folder where the OpenSSL executable is located.
     
     ```
     cd C:\Program Files (x86)\Common Files\MicroStrategy\OpenSSL\openssl-1.0.2e
     ```
  3. Run the OpenSSL executable.
     
     ```
     openssl.exe
     ```
  4. Create the request for a new private key and certificate.
     Following is a sample command that creates a 3072-bit key and a certificate valid for 365 days. If you want a larger key size or a longer certificate lifetime, update the command appropriately.
     
     ```
     ```

- For Linux:
  1. Log into the command line.
  2. Run the OpenSSL utility to create the request for a new private key and certificate. Following is a sample command that creates a 3072-bit key and a certificate valid for 365 days. If
you want a larger key size or a longer certificate lifetime, update the command appropriately:

```
```

---

**Regenerate a new Signing Certificate Authority Certificate**

Follow these steps if you have a Signing CA private key and need to generate a new certificate (for example, your existing CA certificate has expired or is about to expire).

Use OpenSSL to generate a new certificate (.crt) based on your existing private key (.key).

For **Windows**:

1. Open a command prompt window as administrator.

2. Navigate to the folder where the OpenSSL executable is located.

   ```
   > cd C:\Program Files (x86)\Common Files\MicroStrategy\OpenSSL\openssl-1.0.2e
   ```

3. Run the OpenSSL executable.

   ```
   > openssl.exe
   ```

4. Create the request for a new certificate. Following is a sample command that generates a certificate valid for 365 days. If you want a longer certificate lifetime, update the command appropriately.
> req -new -x509 -days 365 -subj
<PathToCertFolder>/ushersigningca.key -out
<PathToCertFolder>/ushersigningca.crt

For Linux:

1. Open a terminal window.

2. Run the OpenSSL utility to create the request for a new certificate. The following is a sample command that generates a certificate valid for 365 days. If you want a longer certificate lifetime, update the command appropriately.

   ```
   > openssl req -new -x509 -days 365 -subj
   <PathToCertFolder>/ushersigningca.key -out
   <PathToCertFolder>/ushersigningca.crt
   ```

Propagate the Signing Certificate Authority

1. Append the Signing CA certificate to the Certificate Authority Chain (.pem) file used by your MicroStrategy Identity environment.

For Windows:

Run the following command:

```
> type "<PathToCertFolder>/ushersigningca.crt"
>> "<PathToCertFolder>/UsherCACChain.pem"
```

For Linux:
Run the following command:

```
> cat <PathToCertFolder>/ushersigningca.crt >
<PathToCertFolder>/UsherCACChain.pem
```

2. To ensure that Tomcat will trust the Signing CA, you must add it to the CA store of the Java Runtime Environment (JRE) used by Tomcat.

For **Windows**:  

1. Run Command Prompt as administrator.
2. Run the following command:

   ```
   "C:\Program Files (x86)\Common Files\MicroStrategy\JRE\180_77\Win64\bin\keytool.exe" -import -alias <NewAlias> -file <PathToCertFolder>/ushersigningca.crt -keystore "C:\Program Files (x86)\Common Files\MicroStrategy\JRE\180_77\Win64\lib\security\cacerts"
   ```

3. Enter the passphrase when prompted (the default is "changeit").

4. Confirm you want to add the certificate to the store when prompted.

For **Linux**:  

1. Log into the command line.
2. Run the following command:

   ```
   keytool -import -alias <NewAlias> -file <PathToCertFolder>/ushersigningca.crt -keystore <PathToJavaRoot>/lib/security/cacerts
   ```

3. Enter the passphrase when prompted (the default is "changeit").

4. Confirm you want to add the certificate to the store when prompted.
Update the MicroStrategy Identity Configuration

1. If you previously completed the MicroStrategy Identity post-installation configuration, and the new Signing CA certificate, or private key, does not have the same file name or path as the old ones you must update those fields during configuration.

2. In a web browser, navigate to MicroStrategy Identity Manager.

3. Use your admin badge to scan the QR code and log in.

4. From the drop-down menu in the upper-right corner, click MicroStrategy Identity Configuration.

5. Update the SSL Certificate Authority Certificate and SSL Certificate Authority Key fields to the new paths.

6. Click Next.

7. If the MicroStrategy Identity Server or MicroStrategy Identity Gateway web services are running, restart them.

8. If you have any apps using MicroStrategy Identity client certificates, you must regenerate those client certificates so they are signed by the new Signing CA and replace them in the apps.
How to Install the Kerberos Authentication Service

1. Install Kerberos KDC server and client.
   a. Download and install the krb5 server package.

   ```
   rpm -ivh krb5-server-1.10.3-10.el6_4.6.x86_64
   ```

   Verify that the following rpm packages are installed before configuring KDC:

   ```
   $ rpm -qa | grep -i krb5
   pam_krb5-2.3.11-9.el6.x86_64
   krb5-server-1.10.3-10.el6_4.6.x86_64
   krb5-workstation-1.10.3-10.el6_4.6.x86_64
   krb5-libs-1.10.3-10.el6_4.6.x86_64
   ```

   b. Install via yum.

   On the KDC server:

   ```
   yum install krb5-server krb5-libs krb5-auth-dialog
   ```

   On the Kerberos client:

   ```
   yum install krb5-workstation krb5-libs krb5-auth-dialog
   ```

2. Modify the `/etc/krb5.conf` file.
Modify `/etc/krb5.conf` to look like the code below with the appropriate REALM and DOMAIN_REALM mappings. `krb5.conf` can be found in `/etc` by default.

```plaintext
[logging]
default = FILE:/var/log/krb5libs.log
dkc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
default_realm = MYREALM.COM
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true

[realms]
MYREALM.COM = {
    kdc = elserver1.example.com
    admin_server = elserver1.example.com
}

[domainrealm]
.myrealm.com = CTCCDH1.COM
myrealm.com = CTCCDH1.COM
```

3. Modify the `KDC.conf` file.

Log in to the KDC server and modify
`/var/kerberos/krb5kdc/kdc.conf` as follows:

```plaintext
[kdcdefaults]
kdc_ports = 88
kdc_tcp_ports = 88

[realms]
MYREALM.COM = {
```
#master_key_type = aes256-cts
acl_file = /var/kerberos/krb5kdc/kadm5.acl
dict_file = /usr/share/dict/words
admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
}

4. Assign administrator privileges.

The users can be granted administrator privileges to the database using the file
/var/kerberos/krb5kdc/kadm5.acl.

*/admin@MYREALM.COM  *

In the above example, any principal in the MYREALM with an admin instance has all administrator privileges.

5. Create a principal.

Create the principal using the command addprinc. For example, create a principal with the user name “root”.

$ kadmin.local -q "addprinc root/admin"

Authenticating as principal root/admin@MYREALM.COM with password.
WARNING: no policy specified for root/admin@MYREALM.COM;
defaulting to no policy
Enter password for principal "root/admin@MYREALM.COM":
Re-enter password for principal "root/admin@MYREALM.COM":
Principal "root/admin@MYREALM.COM" created.

6. Create the database.

The following command creates the principal database in
/var/kerberos/krb5kdc.
The database already exists, it will remove all the related files in /var/kerberos/krb5kdc. By default, the database name is "principal." You can add the -d flag to rename the database.

7. Start the Kerberos Service.

Start the KDC and kadmin daemons as shown below.

```bash
# service krb5kdc start
Starting Kerberos 5 KDC: [ OK ]

# service kadmin start
Starting Kerberos 5 Admin Server: [ OK ]
```
Troubleshooting

This appendix provides information on common problems that you might encounter while installing and configuring MicroStrategy on Linux and Windows operating systems.

Reviewing General Installation Errors

Any errors in your MicroStrategy installation are written to the install.log file.

Review errors found in install.log

1. Browse to INSTALL_PATH where INSTALL_PATH is the directory you specified as the Install Directory during installation.

2. Open the install.log file. (For Unix, use an editor with a command like dtpad install.log. For Windows, use Notepad or some other text editor to open the file.)

3. Review the error messages. A common error is to run out of space.
Graph and Document Support of Non-Western European Fonts

If your Linux system uses non-Western European fonts, you may see indiscernible values returned in place of text on your graphs when accessed through MicroStrategy Web. This also occurs for Report Services documents when accessed through MicroStrategy Web or any other MicroStrategy client application (such as Developer) that connects to an Intelligence Server connected (three-tier) project source.

To support non-Western European fonts, copy True Type fonts into the Intelligence Server installation directory. Copy these fonts, which have a .ttc or .ttf extension, to INTELLIGENCE_SERVER_INSTALL_PATH\PDFGeneratorFiles. The default installation path for the Intelligence Server in Linux is home\MicroStrategy\PDFGeneratorFiles. For the change to take effect, you must restart Intelligence Server.

Server Port Number Errors

I Forgot the Server Port Number

1. Open MicroStrategy Service Manager.
   - **Windows**: From the Start menu, go to Programs > MicroStrategy Tools > Service Manager.
   - **Linux**: In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation. Browse to the folder bin and type ./mstrsvcmgr, then click Enter. MicroStrategy Service Manager opens.
2. Click **Options**.

3. Click **Intelligence Server Options** tab to view the port number.

### Port Number is in Use

In a Linux environment, you can find a port available for use with the following procedure:

1. **Browse to your target directory.** This is the path indicated during installation as the Directory Name.

2. **Browse to the folder** `bin` **and type** `mstrctl -s IntelligenceServer di FindingPortNumber, then click Enter.`

3. **Type** `mstrctl -s IntelligenceServer ci FindingPortNumber, then click Enter.`

4. **Type** `mstrctl -s IntelligenceServer gs FindingPortNumber, then click Enter.`

5. **An XML file is returned.** Search for the tag `<tcp_port_number>`, which contains a port number you can use. Record this number.

### DSN Connection Errors

This section provides troubleshooting information on DSN connection errors.

**Testing the DSN connection failed in DSN Creator**

1. In MicroStrategy Connectivity Wizard, go to the Driver Details page and review all the information.

2. Click each of the boxes and read the comments at the bottom of
3. Enter the information required for each box and click **Test**.

## Metadata and Other Repository Creation Errors

This section provides troubleshooting information on metadata, History List, and statistics repository creation errors.

The errors listed below do not cover errors that can occur because a user does not have the correct database permissions to create a metadata, History List, or statistics repository. For information on providing the proper database permissions to create these repositories, see *Required database permissions to create metadata, History List, and statistics repositories, page 376*.

### Creating a Metadata Fails Due to Insufficient Page Size

When creating a metadata within a IBM DB2 database, the following error and resulting error message can be encountered:

```
Invalid Page Size for DB2 UDB Metadata
```

This can occur when an IBM DB2 database user has access to a tablespace that does not have the space requirements necessary for the metadata repository. The MicroStrategy metadata repository requires a page size of at least 8 KB.

To avoid this error, it is recommended to revoke access to any tablespaces that are lower than 8 KB in page size for the user account that is creating the metadata. The user must also have access to a tablespace with a page size of at least 8 KB. Contact your database administrator to perform this user configuration.
Creating a History List Repository Fails Due to Insufficient Page Size

When creating a History List repository within a Sybase database, the following error and resulting error message can be encountered.

Creating table TableName failed because the minimum row size would be PageSizeRequirement bytes. This exceeds the maximum allowable size of a row for this table, PageSizeLimit bytes.

This can occur when a Sybase database does not have the page size requirements necessary for the History List Repository. To solve this error, contact your database administrator to increase the current available space (PageSizeLimit in the error message above) for the table space to be large enough to store the History List repository (PageSizeRequirement in the error message above). As a general rule, the page size for the History List repository should be at least 4 KB.

Permission Errors

This section provides troubleshooting information on permission errors in a Linux environment.

Missing JVM File

The installation fails just before it starts transferring files, and the following error is displayed:

JVM not found

Clear the set group ID on execution(s) bit on the permissions of the directory where the InstallPath for the Intelligence Server is to be placed.
1. `#chmod g-s directory`

   where *directory* is the InstallPath for Intelligence Server.

2. click **Enter**.