Installation and Configuration Guide
CONTENTS

Book Overview and Additional Resources .......................................................... 1
  About this book .................................................................................................. 1

1. Planning Your Installation .................................................................................. 1
  Upgrade considerations ...................................................................................... 1
  MicroStrategy products and components ............................................................. 1
  Installation prerequisites .................................................................................... 18
  Installation considerations .................................................................................. 35
  Methods of installation ....................................................................................... 45
  Licensing information ......................................................................................... 48
  Installation and configuration checklists ............................................................... 48

2. Installing MicroStrategy on Windows ............................................................... 52
  Installation procedure ......................................................................................... 53
  Configuring your MicroStrategy installation ....................................................... 78

3. Installing MicroStrategy on UNIX and Linux .................................................... 79
  Installation procedures on UNIX and Linux ......................................................... 79
  Usher Security Installation and Configuration .................................................... 102
  Configuring your MicroStrategy installation ....................................................... 120

4. Activating Your Installation .............................................................................. 121
  Request an Activation Code ................................................................................ 121
  Activate your installation .................................................................................... 123
  Configuring your MicroStrategy installation ....................................................... 124
  Server Activation FAQ ....................................................................................... 124

5. Configuring and Connecting Intelligence Server .............................................. 131
  Communicating with databases ........................................................................... 131
Initial MicroStrategy configuration .................................................. 140
Connecting to a data warehouse and other repositories .................... 175
Creating a project ........................................................................... 188
Configuring your MicroStrategy installation ..................................... 189

6. Deploying MicroStrategy Web, Mobile Server, and Operations Manager .................................................................................. 190
   Deploying with IIS (Windows) ......................................................... 191
   General steps to deploy MicroStrategy JSP applications .................. 194
   Deploying with WebLogic and Apache (Solaris) .............................. 198
   Deploying with WebSphere and IBM HTTP Server (AIX) ............... 212
   Deploying with Oracle Glassfish Server (Solaris) ......................... 222
   Deploying with Tomcat (Windows) .................................................. 230
   Deploying with Tomcat (Linux) ...................................................... 236
   Deploying with SAP NetWeaver (Windows) .................................... 241
   Deploying with Oracle 10g (Windows) ........................................... 245
   Deploying with JBoss (Windows) ................................................... 248
   Administering your MicroStrategy Web deployment ....................... 254
   Using absolute paths to share configuration files ......................... 255
   Configuring third-party data sources for importing data .................. 255
   Configuring your MicroStrategy installation .................................... 257

7. Setting Up Documents and HTML Documents .................................. 258
   Prerequisites .................................................................................. 258
   Executing documents and HTML documents in UNIX/Linux .......... 259
   Configuring your MicroStrategy installation ..................................... 264

8. Automated Installation on Windows .................................................. 265
   Installation log file .......................................................................... 266
   Methods of installation ................................................................... 266
   Installing and configuring with a response.ini file ......................... 267
   Silent installation ........................................................................... 315
   Configuring your MicroStrategy installation ..................................... 320

9. Automated Installation on UNIX and Linux ...................................... 321
   Silent installation ........................................................................... 321
   Configuring MicroStrategy in command line mode ....................... 341
   Configuring your MicroStrategy installation ..................................... 342
10. Deploying OEM Applications ................................................................. 343
   Creating DSNs for OEM environments ............................................. 344
   Configuring a MicroStrategy installation ........................................ 345
   Designing a project and reporting environment .............................. 345
   Customizing MicroStrategy Web ...................................................... 346
   Deploying a MicroStrategy OEM application .................................. 346
   Tuning an OEM deployment ............................................................. 351
   Updating OEM applications ............................................................ 352
   Troubleshooting support for MicroStrategy OEM applications ......... 353

   Creating a DSN for a data source .................................................... 354
   Testing ODBC connectivity ............................................................. 355
   Configuring MicroStrategy with a response.ini file ....................... 357
   Configuring and controlling Intelligence Server ............................. 371
   Supporting reserved words and characters .................................... 375
   Configuring your MicroStrategy installation ................................ 376

12. Adding or Removing MicroStrategy Components ............................ 377
   Adding or removing MicroStrategy components on Windows .......... 377
   Re-installing MicroStrategy components on Windows .................... 378
   Uninstalling MicroStrategy components on Windows ..................... 380
   Uninstalling MicroStrategy components on UNIX and Linux ......... 381

13. Export Engine configuration .............................................................. 384
   Installation of Export Engine .......................................................... 384
   Changing the port of the Export Engine ......................................... 384
   Connecting Intelligence Server to a specific Export Engine .......... 385
   Connect the Export Engine to a specific Kafka server ................. 385
   Increase concurrency of Export Engine ......................................... 386

A. Connecting to Databases and Data Sources .................................... 387
   Creating DSNs for specific data sources ....................................... 387
   Creating database connections in Web ......................................... 422
   Configuring ODBC parameters with odbc.ini ................................. 424

B. Installing MicroStrategy Hadoop Gateway ...................................... 425
   Connection Modes ......................................................................... 425
   Constraints ................................................................................... 426
System requirements and supported configurations ........................................426
Prerequisites ..............................................................................................426
Steps to deploy MicroStrategy Hadoop Gateway ........................................427
Import Data from Hadoop .........................................................................433
C. Troubleshooting ....................................................................................434
   Reviewing general installation errors ......................................................434
   Graph and document support of non-Western European fonts ..............434
   Server port number errors ....................................................................435
   DSN connection errors .........................................................................435
   Metadata and other repository creation errors ....................................436
   Permission errors ..................................................................................437
Glossary ....................................................................................................438
Index .........................................................................................................475
Book Overview and Additional Resources

The MicroStrategy Installation and Configuration Guide provides comprehensive information on how to install and configure MicroStrategy software, as well as basic maintenance guidelines. This guide gets you started using the UNIX, Linux, and Windows versions of the MicroStrategy platform.

For UNIX and Linux installations, this guide assumes a basic understanding of how to use UNIX and Linux either from a UNIX/Linux server or by using a viewer on a PC.

For tasks that require advanced system administration commands, this document assumes you are either familiar with those commands or can contact your system administrator.

The main chapters of this guide follow the recommended progression of steps to install and configure MicroStrategy. Refer to Chapter 1, Planning Your Installation for important installation prerequisites before you begin installing MicroStrategy.

For details on how to progress through the remaining chapters of this guide, see the section Installation and configuration checklists, page 48 of Chapter 1, Planning Your Installation.

The appendixes contain the following additional reference information, which you may require depending on your specific needs:

- Connecting to Databases and Data Sources provides details and steps to configure ODBC and DSNs for your data warehouse connections.

- Troubleshooting provides various troubleshooting steps and techniques to take in certain installation and configuration scenarios.

About this book

The chapters in this book provide information about:

- All MicroStrategy components
- Installation and configuration procedures
• System tuning considerations
• Troubleshooting and maintenance guidelines

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

How to find business scenarios and examples

Within this guide, many of the concepts discussed are accompanied by business scenarios or other descriptive examples. Many of the examples use the MicroStrategy Tutorial, which is MicroStrategy's sample warehouse and project. Information about the MicroStrategy Tutorial, which is included as part of the MicroStrategy Analytics Modules, can be found in the Basic Reporting Guide.

Other examples in this book may use the Human Resources Analytics Module project, which includes a set of sample reports and dashboards, and the objects use to build those reports and dashboards.

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

What's new in this guide

MicroStrategy 10.6

• Enhancements to the Export Engine will provide users with high-fidelity export of dashboards to PDF. By default, the Export Engine is installed with the MicroStrategy Enterprise Platform. Optionally, you can install and configure the Export Engine on a different server to level the processing load and improve performance. Additionally, error logging is available for troubleshooting purposes. For more information, see Export Engine configuration.

MicroStrategy 10.5

• On Linux environments, MicroStrategy now includes Messaging Services. This offers a fast and scalable message broker for distributed deployments. For new installations, the Messaging Services is enabled out-of-the-box by default. For upgrades to the 10.5 Feature Release, Messaging Services needs to be configured in conjunction with the existing Intelligence Server(s). For more information, see MicroStrategy Messaging Services.

• MicroStrategy has added support for various drivers. For more information, see the MicroStrategy 10.5 Readme.
**MicroStrategy 10.4**

- When installing a new instance of the MicroStrategy Platform using the Express installation option, you can perform the install by a silent installation, enabling the installation inside custom applications or other installations.

- Installation prerequisites and other important information has been updated to reflect MicroStrategy 10.4. For complete information, see the [MicroStrategy 10.4 Readme](#).

**MicroStrategy 10.3**

- You can install the entire MicroStrategy Enterprise Platform on one Windows server using Express Installation, significantly reducing the time for end-to-end installation and configuration. For steps, see *Performing a MicroStrategy Express installation*.

- Installation prerequisites and other important information has been updated to reflect MicroStrategy 10.3. For complete information, see the [MicroStrategy 10.3 Readme](#).

**MicroStrategy 10.2**

- MicroStrategy 10.2 offers a streamlined Linux installation process for Usher Security and Usher Analytics. The pre-installation and post-installation steps have been updated to reflect important changes to the installation process. For detailed Usher Security installation steps, see *Usher Security Installation and Configuration*.

**MicroStrategy 10**

- MicroStrategy Operations Manager is a web-based administrative tool that allows you to view and monitor all of your MicroStrategy environments in one place. For steps on deploying Operations Manager, see *Chapter 6, Deploying MicroStrategy Web, Mobile Server, and Operations Manager*.

- You can use MicroStrategy Web to import data from various data sources. Some data sources require you to configure a secure connection between your third-party data source and MicroStrategy Web, as described in *Configuring third-party data sources for importing data, page 255*.

- To define a new database connection directly from Web for users to import data from a data source into MicroStrategy, see *Creating database connections in Web, page 422*.

**MicroStrategy 9.5.1**

- You can install MicroStrategy Usher on a Linux machine. For steps to install Usher and the Analytics Platform, see *Installing MicroStrategy on UNIX and Linux*.
MicroStrategy 9.5 (MicroStrategy 9s)

- MicroStrategy 9.5 integrates the MicroStrategy Analytics Platform with MicroStrategy Usher. MicroStrategy Usher is a mobile identity platform for enterprise security. Usher enables users to electronically validate their identity using the Usher app and mobile badge on their smartphone, instead of entering a password, displaying a physical ID card, or using a physical key.

MicroStrategy 9.5 supports Usher as a primary authentication method for logging into a project through MicroStrategy Mobile and MicroStrategy Web. Using QR code generation in MicroStrategy, users can scan the code with Usher on their smartphones and gain access to the MicroStrategy project.

Usher is also supported for two-factor authentication in Web.

MicroStrategy 9.5 also provides support for biometric security, location-based access restrictions, and time-based access restrictions.

To configure your MicroStrategy 9.5 installation, see the help page for MicroStrategy 9.5.

MicroStrategy Analytics Enterprise

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

MicroStrategy 9.4

- Installation prerequisites and other important information has been updated to reflect MicroStrategy 9.4. This information is provided in Chapter 1, Planning Your Installation.


Prerequisites

Before working with this document, you should be familiar with:

- The nature and structure of the data to use for your business intelligence application

- Your system’s configuration, including details such as hardware configuration, installed applications, available memory, and so on

- If installing on UNIX or Linux, the basics of the UNIX or Linux operating system

Who should use this guide

This document is designed for system administrators who install, configure, and maintain MicroStrategy software on the UNIX, Linux, or Windows operating systems.
This document discusses how to perform automated and silent installations. 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.

**Resources**

**Documentation**

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

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

  The steps to access the manuals are described in *Accessing manuals and other documentation sources, page 10.*

  Most of these manuals are also available printed in a bound, soft cover format. To purchase printed manuals, contact your MicroStrategy Account Executive with a purchase order number.

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

**Translations**

Due to translation time, manuals in languages other than English may contain information that is one or more releases behind. You can see the version number on the title page of each manual.

**Finding information**

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

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

For iOS devices, scan the following QR code:

![QR Code for iOS]

For Android devices, scan the following QR code:

![QR Code for Android]

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

Manuals for MicroStrategy overview and evaluation

• *Introduction to MicroStrategy: Evaluation Guide*

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

• *MicroStrategy Evaluation Edition Quick Start Guide*

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

Resources for Security

• *Usher Help*
Steps to setup your Usher Security network, and control access to logical and physical resources.

**Manuals for query, reporting, and analysis**

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

- **MicroStrategy Upgrade Guide**
  Instructions to upgrade existing MicroStrategy products.

- **MicroStrategy Project Design Guide**
  Information to create and modify MicroStrategy projects, and understand facts, attributes, hierarchies, transformations, advanced schemas, and project optimization.

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

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

- **MicroStrategy Operations Manager**
  Instructions for managing, monitoring, and setting alerts for all of your MicroStrategy systems from one console. This guide also includes instructions for setting up and using Enterprise Manager to analyze your MicroStrategy system usage.

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

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

- **MicroStrategy Dashboards and Widgets Creation Guide: Creating Interactive Dashboards for your Data**
  Instructions for designing and creating MicroStrategy Report Services dashboards, a type of document that is optimized for viewing online and for user interactivity. It builds on the basic concepts about documents presented in the MicroStrategy Report Services Document Creation Guide.
• **MicroStrategy In-memory Analytics Guide**

  Information to use MicroStrategy OLAP Services features, including Intelligent Cubes, derived metrics, derived elements, dynamic aggregation, view filters, and dynamic sourcing.

• **MicroStrategy Office User Guide**

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

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

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

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

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

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

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

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

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

• **MicroStrategy Functions Reference**

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

• **MicroStrategy MDX Cube Reporting Guide**

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

**Manuals for Analytics Modules**

• **Manual for the Human Resources Analytics Module**

• **Human Resources Analytics Module Reference**
Software Development Kits

- **MicroStrategy Developer Library (MSDL)**
  Information to understand the MicroStrategy SDK, including details about architecture, object models, customization scenarios, code samples, and so on.

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

Documentation for MicroStrategy Portlets

- **Enterprise Portal Integration Help**
  Information to help you implement and deploy MicroStrategy BI within your enterprise portal, including instructions for installing and configuring out-of-the-box MicroStrategy Portlets for several major enterprise portal servers.
  This resource can be accessed from the MicroStrategy Product Manuals page, as described in *Accessing manuals and other documentation sources, page 10.*

Documentation for MicroStrategy GIS Connectors

- **GIS Integration Help**
  Information to help you integrate MicroStrategy with Geospatial Information Systems (GIS), including specific examples for integrating with various third-party mapping services.
  This resource can be accessed from the MicroStrategy Product Manuals page, as described in *Accessing manuals and other documentation sources, page 10.*

Help

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

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

MicroStrategy provides several ways to access help:

- Help button: Use the Help button or ? (question mark) icon on most software windows to see help for that window.

- Help menu: From the Help menu or link at the top of any screen, select MicroStrategy Help to see the table of contents, the Search field, and the index for the help system.
• F1 key: Press F1 to see context-sensitive help that describes each option in the software window you are currently viewing.

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

Accessing manuals and other documentation sources

The manuals are available from https://microstrategyhelp.atlassian.net, as well as from the machine where MicroStrategy was installed.

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

The best place for all users to begin is with the MicroStrategy Basic Reporting Guide. To access the installed manuals and other documentation sources, see the following procedures:

• To access documentation resources from any location, page 10
• To access documentation resources on Windows, page 10
• To access documentation resources on UNIX and Linux, page 11

To access documentation resources from any location

1 Visit https://microstrategyhelp.atlassian.net/.

To access documentation resources on Windows

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

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

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

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

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

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

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

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

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

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

Documentation standards

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

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>• Button names, check boxes, options, lists, and menus that are the focus of actions or part of a list of such GUI elements and their definitions</td>
</tr>
<tr>
<td></td>
<td>Example: Click <strong>Select Warehouse</strong>.</td>
</tr>
<tr>
<td><strong>italic</strong></td>
<td>• Names of other product manuals and documentation resources</td>
</tr>
<tr>
<td></td>
<td>• When part of a command syntax, indicates variable information to be replaced by the user</td>
</tr>
<tr>
<td></td>
<td>Example: The <strong>aggregation level</strong> is the level of calculation for the metric.</td>
</tr>
<tr>
<td></td>
<td>Example: Type <code>copy c:\filename d:\foldername\filename</code></td>
</tr>
<tr>
<td><strong>Courier font</strong></td>
<td>• Calculations</td>
</tr>
<tr>
<td></td>
<td>• Code samples</td>
</tr>
<tr>
<td>Type</td>
<td>Indicates</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
</tr>
<tr>
<td>•</td>
<td>Registry keys</td>
</tr>
<tr>
<td>•</td>
<td>Path and file names</td>
</tr>
<tr>
<td>•</td>
<td>URLs</td>
</tr>
<tr>
<td>•</td>
<td>Messages displayed in the screen</td>
</tr>
<tr>
<td>•</td>
<td>Text to be entered by the user</td>
</tr>
</tbody>
</table>

**Example:** \( \text{Sum(revenue)}/\text{number of months} \).

**Example:** Type `cmdmgr -f scriptfile.scp` and press **Enter**.

| +    | A keyboard command that calls for the use of more than one key (for example, `SHIFT+F1`). |

| ☰    | A note icon indicates helpful information for specific situations. |

| 🔴    | A warning icon alerts you to important information such as potential security risks; these should be read before continuing. |

---

**Education**

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

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

- MicroStrategy Developer: Reporting Essentials
- MicroStrategy Web: Report Analysis
- MicroStrategy Web: Report Design

For a detailed description of education offerings and course curriculums, visit [www.microstrategy.com/Education](http://www.microstrategy.com/Education).
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 have earlier versions of MicroStrategy products installed and want to upgrade them, refer to the Upgrade Guide. The upgrade procedures on Windows, UNIX, and Linux are covered in this guide. It is recommended that you read this guide before deciding to upgrade an 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.

**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, UNIX, and Linux environments. For information on how to deploy MicroStrategy Web (ASP.NET) and MicroStrategy Web (JSP), see *Chapter 6, Deploying MicroStrategy Web, Mobile Server, and Operations Manager*.

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**

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

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.


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 5.

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.

To allow users to install MicroStrategy Office from a network location

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 MicroStrategyOffice.msi or MicroStrategyOffice64.msi files to perform a silent installation of MicroStrategy Office (see Silent installation of MicroStrategy Office, page 318).
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 UNIX, 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 14 and the accompanying MSDL.

For information on deploying MicroStrategy Web Services ASP.NET and J2EE versions, refer to the MicroStrategy Office User Guide.

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 Design and Administration Guide](#) and the [MicroStrategy Mobile Analysis Guide](#).

For information on how to deploy MicroStrategy Mobile Server (ASP.NET) and MicroStrategy Mobile Server (JSP), see Chapter 6, *Deploying MicroStrategy Web, Mobile Server, and Operations Manager*.

## 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 7**
- **MicroStrategy Report Services, page 7**
- **MicroStrategy Distribution Services, page 8**
- **MicroStrategy Transaction Services, page 9**
- **MultiSource Option, page 9**
- 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, dashboards, 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 Operations 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 Operations Manager**

MicroStrategy Operations Manager is a web-based administrative tool that allows you to view and monitor all your MicroStrategy environments in one place by pulling information from the MicroStrategy Health Center system in each of your environments.

For information on using Operations Manager, see the [Operations Manager 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 8*.

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 UNIX or Linux machine (see *MicroStrategy ODBC Driver for SequeLink, page 398*).
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.

Customers interested in deploying analytics from the R programming language into MicroStrategy can do so using the R Integration Pack, available separately from the CodePlex open source community web site. For more information, go to [http://www.codeplex.com](http://www.codeplex.com) and search for the **RIntegrationPack** project (current as of March 1, 2013).
• 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 that are designed to allow users to customize and extend MicroStrategy products and to integrate them 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 robust functionality of 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 from the MicroStrategy support site [https://resource.microstrategy.com/msdz/default.asp](https://resource.microstrategy.com/msdz/default.asp). You can also access the MicroStrategy Developer Library from the MicroStrategy support site.
The MicroStrategy SDK provides access to the entire MicroStrategy platform and includes all of the services and utilities required for building a robust, 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
  - Visualization SDK
  - Mobile SDK
  - Web Services 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:
    - COM-based client-server API
    - XML-based Web API with support for Java/COM
    - Web Services API
    - 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 that reduce code creation and 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 dashboards 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 dashboards 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 dashboards, 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 Health Center**

MicroStrategy Health Center can help you diagnose and fix problems in your MicroStrategy system. It detects known problems and provides an immediate solution. In cases where Health Center cannot fix a problem immediately, it enables you to bundle relevant log files into a diagnostic package and transmit this package to MicroStrategy Technical Support for review and troubleshooting.

Health Center is provided with a MicroStrategy installation.

For information on using Health Center to diagnose and fix problems in your MicroStrategy environment, see the Project Design Guide.

**MicroStrategy Usher**

MicroStrategy Usher is a mobile security platform designed to provide security for business processes and applications across an enterprise. It replaces traditional forms of enterprise security such as ID cards, passwords, and physical keys with a mobile badge on a user’s smartphone. Users with the Usher Security mobile app and badge can
electronically validate their identity without plastic ID cards, log in to online applications without entering a password, open locked doors without keys, and so on.

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

- **Usher Security Server**: Server system that synchronizes identities with enterprise identity management systems of record, and presents those identities to Usher clients for authentication.

- **Usher Network Manager**: Administrative console that allows you to manage your network of users, configure access to Usher-enabled systems and resources, and distribute digital badges and keys.

- **Usher Analytics**: Reporting functionality to analyze and visualize the activity of users in your Usher network to gain insights into your enterprise security.

- **Usher Professional**: Mobile app and server system that allows your users to view the activity of nearby users in their Usher network.

Each user downloads Usher Security to their mobile device. Usher Security is a mobile app that allows users to validate their identities or to access Usher-enabled systems and resources.

For installation instructions, refer to the *Usher Security Installation and Configuration*.

### 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 46.

- 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.

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

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

- *Recommended installation location and example deployments, page 19*
- *Hardware requirements and recommendations, page 22*
- *Software requirements and recommendations, page 29*

### 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
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 *MicroStrategy Readme* on the MicroStrategy website.

For UNIX and 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 UNIX and Linux, page 43*.

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 hotfix 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>MicroStrategy Product</th>
<th>Processor</th>
<th>Memory</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroStrategy System Manager</td>
<td>x86 or x64</td>
<td>2 GB</td>
<td>0.5 GB</td>
</tr>
<tr>
<td></td>
<td>compatible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Command Manager</td>
<td>x86 or x64</td>
<td>2 GB</td>
<td>0.25 GB</td>
</tr>
<tr>
<td></td>
<td>compatible</td>
<td></td>
<td></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</td>
<td>1 GB</td>
<td>0.25 GB</td>
</tr>
<tr>
<td></td>
<td>compatible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Developer products</td>
<td>x86 or x64</td>
<td>2 GB or higher</td>
<td>0.25 GB</td>
</tr>
<tr>
<td></td>
<td>compatible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Product</td>
<td>Processor</td>
<td>Memory</td>
<td>Storage</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>MicroStrategy Intelligence Server</td>
<td>x64 compatible</td>
<td>4 GB or higher</td>
<td>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.</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 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 Services, and to support optimal performance for MicroStrategy Report Services documents and dashboards and the other features of the MicroStrategy Product Suite.
<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>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 Narrowcast Server</td>
<td>x86 or x64 compatible</td>
<td>4 GB</td>
<td>4 GB</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 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>MicroStrategy Web can be accessed from a third-party web browser, which means there are no additional storage requirements.</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 UNIX/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.
Be aware of the following:

- 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 *Storage requirements for all MicroStrategy products on UNIX/Linux, page 28.*

- A successful configuration of Intelligence Server depends on a valid combination of an operating system and a CPU architecture. For example, Intelligence Server is certified to run on a machine with an Oracle Solaris operating system compatible with a SPARC 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>Oracle Solaris: SPARC platforms</td>
<td>2 GB or higher</td>
<td>• 4 GB on IBM AIX</td>
</tr>
<tr>
<td></td>
<td>IBM AIX: IBM POWER Architecture</td>
<td></td>
<td>• 3 GB on other UNIX operating systems</td>
</tr>
<tr>
<td></td>
<td>Red Hat and SUSE Linux: x86-64 compatible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Command Manager</td>
<td>Oracle Solaris: SPARC platforms</td>
<td>2 GB or higher</td>
<td>• 4 GB on IBM AIX</td>
</tr>
<tr>
<td></td>
<td>IBM AIX: IBM POWER Architecture</td>
<td></td>
<td>• 3 GB on other UNIX operating systems</td>
</tr>
<tr>
<td></td>
<td>Red Hat and SUSE Linux: x86-64 compatible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Intelligence Server</td>
<td>Oracle Solaris: SPARC platforms</td>
<td>4 GB or higher</td>
<td>Three times the amount of RAM available to Intelligence Server. For example, an</td>
</tr>
<tr>
<td></td>
<td>Using 4 GB of RAM is a minimum level of support for the MicroStrategy</td>
<td></td>
<td></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></td>
<td></td>
<td>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 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 Services, and to support optimal performance for MicroStrategy Report Services documents and dashboards and the other features of the MicroStrategy Product Suite.</td>
<td>Intelligence Server that is provided 4 GB of RAM requires 12 GB of hard drive space.</td>
</tr>
<tr>
<td></td>
<td>IBM AIX: IBM POWER Architecture</td>
<td>2 GB or higher</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Red Hat and SUSE Linux: x86-64 compatible</td>
<td>2 GB or higher</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Integrity Manager</td>
<td>Oracle Solaris: SPARC platforms</td>
<td>2 GB or higher</td>
<td>4 GB on IBM AIX 3 GB on other UNIX operating systems</td>
</tr>
<tr>
<td></td>
<td>IBM AIX: IBM POWER Architecture</td>
<td>2 GB or higher</td>
<td>4 GB on IBM AIX 3 GB on other UNIX operating systems</td>
</tr>
<tr>
<td></td>
<td>Red Hat and SUSE Linux: x86-64 compatible</td>
<td>2 GB or higher</td>
<td>4 GB on IBM AIX 3 GB on other UNIX operating systems</td>
</tr>
<tr>
<td>MicroStrategy Web Services for Office</td>
<td>Oracle Solaris: SPARC platforms</td>
<td>2 GB or higher</td>
<td>4 GB on IBM AIX 3 GB on other UNIX operating systems</td>
</tr>
<tr>
<td></td>
<td>IBM AIX: IBM POWER Architecture</td>
<td>2 GB or higher</td>
<td>4 GB on IBM AIX 3 GB on other UNIX operating systems</td>
</tr>
<tr>
<td></td>
<td>Red Hat and SUSE Linux: x86-64 compatible</td>
<td>2 GB or higher</td>
<td>4 GB on IBM AIX 3 GB on other UNIX operating systems</td>
</tr>
</tbody>
</table>
MicroStrategy Mobile Server

The MicroStrategy Mobile Server hardware requirements are the same as those for MicroStrategy Web Server hardware requirements.

MicroStrategy SDK

The SDK is not included in the MicroStrategy installation and can instead be downloaded from the MicroStrategy support site.

MicroStrategy Web: Web Server

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Total Installation Size for All Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX</td>
<td>7.5 GB</td>
</tr>
<tr>
<td>Solaris</td>
<td>8 GB</td>
</tr>
<tr>
<td>Linux</td>
<td>8 GB</td>
</tr>
</tbody>
</table>

MicroStrategy Web: Web Client

Oracle Solaris: SPARC platforms

IBM AIX: IBM POWER Architecture

Red Hat and SUSE Linux: x86-64 compatible

4 GB or higher

- 4 GB on IBM AIX
- 3 GB on other UNIX operating systems

MicroStrategy Web can be accessed from a third-party web browser, which means there are no additional storage requirements.

Storage requirements for all MicroStrategy products on UNIX/Linux

The storage recommendations listed in System hardware requirements and recommendations on UNIX/Linux, page 25 provide the storage recommendations for the MicroStrategy products if they are installed individually on separate machines. The table below lists 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:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Total Installation Size for All Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX</td>
<td>7.5 GB</td>
</tr>
<tr>
<td>Solaris</td>
<td>8 GB</td>
</tr>
<tr>
<td>Linux</td>
<td>8 GB</td>
</tr>
</tbody>
</table>

Temporary directory requirements for installation

In addition to the space requirements listed in System hardware requirements and recommendations on UNIX/Linux, page 25, you also need free space in the temporary directory. The default location of the temporary directory is:

- Solaris: /var/tmp
• AIX and Linux: /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 *MicroStrategy 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 *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 UNIX/Linux**

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

MicroStrategy Integrity Manager for UNIX and 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 UNIX and Linux platforms.
The following MicroStrategy products require an X-windows-enabled environment on all UNIX and Linux platforms:

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

The following requirements also apply to all UNIX and Linux platforms:

- A Web browser (for example, Firefox) is required for viewing the MicroStrategy Readmes and online help.

The requirements listed below describe general requirements as well as requirements specific to the UNIX and Linux platforms:

- Configuring shared memory resources, page 31
- Oracle Solaris, page 32
- Red Hat and SUSE Linux, page 32

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 UNIX and 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 (on the System Requirements page, see Installing with the MicroStrategy Installation Wizard, page 83), 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 provided 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 UNIX and 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 UNIX and Linux documentation and contact your system administrator to modify these settings.

Shared memory settings on UNIX and Linux operating systems that may require modification to support Intelligence Server execution are listed in the table below:

<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Description</th>
<th>Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>shmmni</td>
<td>Maximum number of shared memory identifiers at any given time.</td>
<td>4096</td>
</tr>
<tr>
<td>shmseg</td>
<td>Maximum number of segments per process.</td>
<td>2048</td>
</tr>
</tbody>
</table>

**Note:** This setting does not exist on Solaris, AIX, or Linux operating systems.
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 Solaris and AIX</th>
<th>Description</th>
<th>Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>semmni</td>
<td>Maximum number of system wide semaphore sets.</td>
<td>4096</td>
</tr>
<tr>
<td>semmens</td>
<td>Maximum number of semaphores in the system.</td>
<td>4096</td>
</tr>
</tbody>
</table>

<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>semmens</td>
<td>Maximum number of semaphores in the system.</td>
<td>32000</td>
</tr>
<tr>
<td>semopm</td>
<td>Maximum number of operations in a simple semaphore call.</td>
<td>32</td>
</tr>
<tr>
<td>semmni</td>
<td>Maximum number of semaphore sets.</td>
<td>4096</td>
</tr>
</tbody>
</table>

**Oracle Solaris**

For Oracle Solaris requirements, see the System Requirements in the MicroStrategy Readme

**IBM AIX**

For IBM AIX requirements, see the System Requirements in the MicroStrategy Readme

**Red Hat and SUSE Linux**

For Red Hat Enterprise and SUSE 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 UNIX/Linux, page 29*.

**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 ASP.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 33*).

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. The Windows Features dialog box opens.
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 IHttp Features**, and select:

- **Default Document**
- **Static Content**

d  Expand **Security**, and select:

- **Request Filtering**
- **Windows Authentication**

5  Click **OK** to save your changes.

# 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 in to 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 simply 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 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](http://www.microstrategy.com/services-support/overview). 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](http://www.microstrategy.com/services-support/overview).

**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](http://www.microstrategy.com/services-support/overview).

**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](http://www.microstrategy.com/services-support/overview).
**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 been installed before</td>
<td>The MicroStrategy Installation Wizard prompts you to select the language from the drop-down list. 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 application has been installed before</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. The user language in the product interface is also the language that you selected the first time you installed the product on the system.</td>
</tr>
<tr>
<td>Completely uninstalling all the MicroStrategy products and installing the same version or a newer version</td>
<td>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. Note: 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 8.0.2 can only connect to Intelligence Server 8.0.0 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, Solaris, AIX, HP-UX, and Redhat or SUSE 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 *MicroStrategy Readme*.

**Certificates used during Usher Installation and Configuration**

<table>
<thead>
<tr>
<th>Name</th>
<th>Example file path</th>
<th>When to use</th>
<th>What the certificate is used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usher Server SSL/HTTPS Certificate</td>
<td>/certs/f1_usherservercert.crt</td>
<td>Usher Installation, step 12: field 1</td>
<td>SSL Certificate used by the Tomcat instance hosting the Usher Server</td>
</tr>
<tr>
<td>Usher Server SSL/HTTPS Certificate private key</td>
<td>/certs/f2_usherservercert.key</td>
<td>Usher Installation, step 12: field 2</td>
<td>Private key for the SSL Certificate used by the Tomcat instance hosting the Usher Server</td>
</tr>
<tr>
<td>Usher Server Certificate Trust Bundle</td>
<td>/certs/f3_usherserverbundle.pem</td>
<td>Usher Installation, step 12: field 3</td>
<td>Certificate store that includes all the CA certificates that the Usher Server will trust. File of concatenated PEM-encoded CA Certificates for Client Auth.</td>
</tr>
<tr>
<td>Usher Server Signing Certificate Authority Certificate</td>
<td>/certs/f9_ushersigningcertificate.crt</td>
<td>Setting up Network Manager: providing company information, step 4</td>
<td>Certificate that will be used by the Usher Server to sign any Certificate Signing Request (Directory Agent, Usher Security apps, Usher SDK-based applications, VPN Agents, etc.). It is used by Usher internally only.</td>
</tr>
<tr>
<td>Usher Server Signing Certificate Authority Certificate private key</td>
<td>/certs/f10_ushersigningcertificate.key</td>
<td>Setting up Network Manager: providing company information, step 4</td>
<td>Key for the Usher Signing Certificate</td>
</tr>
<tr>
<td>Usher SAML Certificate</td>
<td>/certs/f11_ushersamlcertificate.crt</td>
<td>Setting up Network Manager: providing company</td>
<td>SAML Certificate that the Usher Server will use to sign SAML Requests</td>
</tr>
<tr>
<td>Name</td>
<td>Example file path</td>
<td>When to use</td>
<td>What the certificate is used for</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Usher SAML Certificate Private Key</td>
<td>/certs/f11_ushersamlcertificate.key</td>
<td>Setting up Network Manager: providing company information, step 4</td>
<td>Private key for the SAML Certificate that the Usher Server will use to sign SAML Requests</td>
</tr>
<tr>
<td>Usher Agent Certificate</td>
<td>/certs/f13_usherdirectoryagentcertificate.crt</td>
<td>Synchronizing users from Active Directory, step 7</td>
<td>SSL Client Certificate the Usher Directory Agent will use to authenticate itself with the Usher Server and the Usher Gateway Server. It is signed by the Usher Signing Certificate through the Agent setup process on Network Manager.</td>
</tr>
<tr>
<td>Usher Agent Certificate Private Key</td>
<td>/certs/f14_usherdirectoryagentcertificate.key</td>
<td>Synchronizing users from Active Directory, step 8</td>
<td>Private key for the Usher Agent Certificate</td>
</tr>
</tbody>
</table>

**Recommended system settings for UNIX and Linux**

UNIX and 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 UNIX and 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 UNIX and 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 (limit name)</td>
<td>Maximum CPU seconds per process</td>
<td>Unlimited</td>
</tr>
<tr>
<td>cputime</td>
<td>Maximum CPU seconds per process</td>
<td>Unlimited</td>
</tr>
<tr>
<td>filesize (file)</td>
<td>Maximum size for a single file</td>
<td>Unlimited, or as large as the file system allows. Your system administrator may enforce limits on the maximum size of files for reasons unrelated to MicroStrategy. This value must be at least as large as the maximum size for core dump files (coredumpsize). AIX machines use a default filesize limit of 2 GB. To support copying the MicroStrategy installation file to an AIX machine, you may need to increase the filesize limit to be larger than the size of the MicroStrategy installation file.</td>
</tr>
<tr>
<td>datasize (data)</td>
<td>Maximum heap size per process</td>
<td>Unlimited, or as large as the system virtual memory allows. Your system's virtual memory constraints affect the data size you can set for a process's heap size. The value should be the same as the maximum size for core dump files (coredumpsize).</td>
</tr>
<tr>
<td>stacksize (stack)</td>
<td>Maximum stack size per process</td>
<td>200 MB</td>
</tr>
<tr>
<td>coredumpsize (coredump)</td>
<td>Maximum size for a single core dump file</td>
<td>Set this value to the same value as the maximum heap size per process (datasize). If core dump files are created that are larger than this value, the files are corrupted and unusable.</td>
</tr>
<tr>
<td>memoryuse (memory)</td>
<td>Maximum size of physical memory allotted per process</td>
<td>Unlimited, or as large as the physical memory of your system allows</td>
</tr>
<tr>
<td>vmemoryuse (vmemory)</td>
<td>Maximum size of virtual memory allowed per process</td>
<td>Unlimited, or as large as your system virtual memory allows</td>
</tr>
</tbody>
</table>
### Setting Name

<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Description</th>
<th>Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ulimit name (limit name)</code></td>
<td><code>descriptors (nofiles)</code></td>
<td>Maximum number of file descriptors (open files) per process</td>
</tr>
<tr>
<td></td>
<td><code>processes (per user)</code></td>
<td>Maximum number of processes per user</td>
</tr>
</tbody>
</table>

This setting is a general guideline that has been observed to work well on multiple installs. However, depending on the specific environment, this setting may need to be refined.

### Enhancements to the Export Engine

Enhancements have been made to the Export Engine for handling of Visual Insight dashboards. The Export Engine is used by Intelligence Server to create PDF files from documents, reports, and Visual Insight dashboards. To view the steps necessary to configure your Export Engine settings, see *Export Engine configuration*.

### Methods of installation

The methods of MicroStrategy installation are:

#### Graphical user interface

The GUI mode presents a user interface for each page in the MicroStrategy Installation Wizard. You click the mouse to place the cursor on the desired object, then proceed as appropriate to complete the task. The following navigational buttons are also displayed:

- **Next**: Click to proceed to the next page.
- **Back**: Click to return to the previous page.
- **Cancel**: Click to cancel the installation and close the MicroStrategy Installation Wizard.
- **Finish** (only on the MicroStrategy Installation Wizard Complete page): Click to complete the setup and close the wizard.

#### 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]. Press 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 press ENTER to proceed to the next page.
• Press 2 and then press ENTER to return to the previous page.
• Press 3 and then press 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 press ENTER to complete the setup and close the wizard.

For information on command line installation, refer to Chapter 3, Installing MicroStrategy on UNIX and 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 315 in Chapter 8, Automated Installation on Windows and Silent installation, page 321 in Chapter 9, Automated Installation on UNIX and 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 hotfix installation, you must include all of the files provided as part of the hotfix installation in their default location. This means that you cannot use the steps provided below to create a custom MicroStrategy hotfix 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 *To determine the required installation files for Windows installations, page 47*.
   - For UNIX and 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 *To specify the location of the installation files, page 48* 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>
</tbody>
</table>

*To determine the required installation files for Windows installations*

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** to close the MicroStrategy Installation Wizard.

**To specify the location of the installation files**

9  When performing the installation, you can specify the location of these files as follows:

- For Windows installations, you can specify the location of these files using a *response.ini* file, as described in *Installation Files, page 281*.
- For UNIX installations, you can specify the location of these files using an *options.txt* file, as described in *Providing installation files for smaller installations, page 339*.

**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 Usher 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 such Usher functionality 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, UNIX, 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 below. 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 chapters sequentially.

You can use the Complete column on the left to check off each high-level step as you complete it.

<table>
<thead>
<tr>
<th>Complete</th>
<th>Chapter and Installation Task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Chapter 1, Planning Your Installation:</em> Review this chapter for important installation prerequisites and considerations.</td>
</tr>
<tr>
<td></td>
<td><em>Chapter 2, Installing MicroStrategy on Windows:</em> This chapter describes the procedures for installing the MicroStrategy products necessary to run your business intelligence application in a Windows environment. Or</td>
</tr>
<tr>
<td></td>
<td><em>Chapter 8, Automated Installation on Windows:</em> 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, <em>Chapter 10, Deploying OEM Applications</em> explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.</td>
</tr>
<tr>
<td></td>
<td><em>Chapter 4, Activating Your Installation:</em> 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.</td>
</tr>
<tr>
<td></td>
<td><em>Chapter 5, Configuring and Connecting Intelligence Server:</em> 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.</td>
</tr>
<tr>
<td></td>
<td><em>Chapter 6, Deploying MicroStrategy Web, Mobile Server, and Operations Manager:</em> 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, UNIX, and Linux platforms with various Web and application servers.</td>
</tr>
<tr>
<td></td>
<td><em>Chapter 12, Adding or Removing MicroStrategy Components:</em> 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:</td>
</tr>
<tr>
<td></td>
<td>• <em>Adding or removing MicroStrategy components on Windows, page 377.</em></td>
</tr>
<tr>
<td></td>
<td>• <em>Re-installing MicroStrategy components on Windows, page 378.</em></td>
</tr>
<tr>
<td></td>
<td>• <em>Uninstalling MicroStrategy components on Windows, page 380.</em></td>
</tr>
</tbody>
</table>
Installing and configuring MicroStrategy on UNIX and Linux

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

You can use the Complete column on the left to check off each high-level step as you complete it.

<table>
<thead>
<tr>
<th>Complete</th>
<th>Chapter and Installation Task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Chapter 1, Planning Your Installation:</strong> Review this chapter for important installation prerequisites and considerations.</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 3, Installing MicroStrategy on UNIX and Linux:</strong> This chapter describes the procedures for installing the MicroStrategy products necessary to run your business intelligence application on a UNIX and Linux environment. Or <strong>Chapter 9, Automated Installation on UNIX and Linux:</strong> 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 UNIX. Or <strong>Usher Security Installation and Configuration:</strong> This chapter describes the full procedures for installing Usher Security in conjunction with other MicroStrategy products. Additionally, <strong>Chapter 10, Deploying OEM Applications</strong> explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 4, Activating Your Installation:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 5, Configuring and Connecting Intelligence Server:</strong> 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 <strong>Chapter 11, Configuring MicroStrategy Using Command Line Tools:</strong> MicroStrategy tools are provided in command line mode on UNIX and 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.</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 6, Deploying MicroStrategy Web, Mobile Server, and Operations Manager:</strong> 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, UNIX, and Linux platforms with various Web and application servers.</td>
</tr>
<tr>
<td>Complete</td>
<td>Chapter and Installation Task</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 7, Setting Up Documents and HTML Documents</strong>: 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 UNIX or Linux environment.</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 12, Adding or Removing MicroStrategy Components</strong>: This chapter describes the steps to add and remove MicroStrategy components on all supported operating systems. For UNIX and Linux platforms, refer to the following section:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Uninstalling MicroStrategy components on UNIX and Linux, page 381</strong></td>
</tr>
</tbody>
</table>
This chapter 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, you should refer to 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 UNIX/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 chapter has the following sections:

- **Installation procedure, page 53**
- **Configuring your MicroStrategy installation, page 78**

If you are installing MicroStrategy on UNIX or Linux, refer to Installation procedures on UNIX and Linux, page 79 in Chapter 3, Installing MicroStrategy on UNIX and Linux.
Additionally, Chapter 10, Deploying OEM Applications explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.

**Installation procedure**

The MicroStrategy Installation Wizard provides steps to guide 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, page 53*
- *Installation verification, page 68*

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

- For information on installation prerequisites, see *Installation prerequisites, page 18* in Chapter 1, Planning Your Installation.

- For information about advanced installation functionality, such as installing in an SMS environment or using installation response files, see *Chapter 8, Automated Installation on Windows*.

- For information about installing and deploying MicroStrategy Web with other Web and application servers, see *Chapter 6, Deploying MicroStrategy Web, Mobile Server, and Operations Manager*.

- For information about deploying MicroStrategy Web Services ASP.NET and J2EE, see the *MicroStrategy Office User Guide*.

- For information about installation prerequisites and procedures for Usher Security, see *Usher Security Installation and Configuration*.

**Note the following:**

- 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.

  **i**

- 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**

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**.
Prerequisites

The following prerequisites are in addition to those listed in *Installation prerequisites, page 18:*

- **Usher Security requires a secure web server, contact your IT administrator for assistance.** The following items are needed to successfully configure and access Usher Security services:
  - An SSL Certificate Authority Root certificate and Intermediate certificates, appended into a (.pem) file. You need to obtain this from a third-party signing authority, such as Verisign or Thawte. Contact your IT administrator for assistance.
  - A HTTPS/SSL server certificate (.crt) file, and the matching key (.key) file, which matches your Fully Qualified Domain Name.
  - Connectivity information to a SMTP Server, and the port and credentials to allow the Usher server to send invitation emails.
  - Your Windows computer hostname has been changed to a Fully Qualified Domain Name, and registered in the Domain Name System (DNS).

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 69.*
   - 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 46.* Details on using a `response.ini` file to provide the location of the
installation files are provided in *Chapter 8, Automated Installation on Windows* and the parameters used to specify the location of the required installation files are described in *Installation Files, page 281*.

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 leads 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**

Read the information on the welcome screen and proceed to the next step.

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 12, 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.

- **Click Print** to print a copy of the license agreement for your records.

**Customer Information**

Enter the following customer information:

- **First Name**
- **Last Name**
- **Email Address**
- **License Key**

- **Licensed users can contact Technical Support to obtain a license key.**
**Install Options**

Select one of the following install options:

- To easily 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**

You can select both the MicroStrategy products to install and their final location.

Once you have selected all required MicroStrategy products and defined the proper installation locations, if you are prompted to stop your MicroStrategy Web server, click **Yes**. If you click **No**, you cannot continue with the installation until you stop your MicroStrategy Web server.

**Installing MicroStrategy products**

Select the check box next to 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 selected to install. These instructions describe all possible pages to support all products of the MicroStrategy Product Suite. You do not have to install all of these products on the same machine. In fact, this is strongly discouraged in a production environment. For basic guidelines about product deployments, see Recommended installation location and example deployments, page 19 in Chapter 1, Planning Your Installation.

Depending on your license key, you can install the platform independent version of some of the products listed below.

- MicroStrategy Intelligence Server (see MicroStrategy Intelligence Server, page 6)
- MicroStrategy Web (see MicroStrategy Web components, page 2)
  - MicroStrategy Portlets (see MicroStrategy Portlets, page 3)
  - MicroStrategy GIS Connectors (see MicroStrategy GIS Connectors, page 3)
- MicroStrategy Operations Manager (see Chapter 6, Deploying MicroStrategy Web, Mobile Server, and Operations Manager)
- MicroStrategy Office (see MicroStrategy Office, page 3)
  - MicroStrategy Web Services for Office (see MicroStrategy Web Services (ASP.NET) and Web Services (J2EE), page 5)

If you are a MicroStrategy Web administrator, you can allow Web users to install MicroStrategy Office by making an ‘Install MicroStrategy Office’ link available in MicroStrategy Web. When a user chooses to install MicroStrategy Office, MicroStrategy Office is installed as a stand-alone product on his or her machine. MicroStrategy Office can be installed even if no other MicroStrategy products are available on his or her machine. For steps to enable users to install MicroStrategy Office from Web, see Enabling users to install MicroStrategy Office from Web, page 254 of Chapter 6, Deploying MicroStrategy Web, Mobile Server, and Operations Manager.

- MicroStrategy Mobile (see MicroStrategy Mobile, page 5)
- MicroStrategy Developer Products (see MicroStrategy Developer, page 12)
- MicroStrategy Object Manager (see MicroStrategy Object Manager, page 14)
- MicroStrategy Command Manager (see MicroStrategy Command Manager, page 9)
- MicroStrategy Enterprise Manager (see MicroStrategy Enterprise Manager, page 10)
- MicroStrategy Integrity Manager (see MicroStrategy Integrity Manager, page 13)
- MicroStrategy System Manager (see MicroStrategy System Manager, page 11)
- MicroStrategy Narrowcast Server (see MicroStrategy Narrowcast Server, page 11)
• MicroStrategy Analytics Modules (see MicroStrategy sample projects, page 16)
• Usher Security (see Usher Help)

   This also includes MicroStrategy Tutorial Reporting (see MicroStrategy Tutorial Reporting, page 16)

• Other components:
  ▪ 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 UNIX or Linux machine (see MicroStrategy ODBC Driver for SequeLink, page 398).

    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.

  ▪ The MicroStrategy MDX Cube Provider is required to connect to IBM Cognos TM1 data sources. It can also be used to connect to Microsoft Analysis Services data sources. For information on connecting to these MDX Cube data sources, see the MDX Cube Reporting Guide.

    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 https://resource.microstrategy.com/msdz/default.asp. You can also access the MicroStrategy Developer Library from the MicroStrategy support site.

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 1 in Chapter 1, Planning Your Installation.

**Defining the installation location for MicroStrategy products and subcomponents**

You can select MicroStrategy products and their sub-components to define their installation locations. When you select a MicroStrategy product or sub-component, 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 sub-component and the Browse button is not accessible, this means that the installation location cannot be changed. For example, if you select MicroStrategy Office 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 56).

- 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 8, Automated Installation on Windows and the parameters used to specify the location of the required installation files are described in Installation Files, page 281.

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.

Usher Configuration

You see this page if you have selected to install Usher. If you do not have all the information and want to configure Usher 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 Usher 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 Usher network.
- **Host Name**: Enter the Fully Qualified Domain Name you are using, for example, yourFQDN.com.

**MicroStrategy Health Center Setting**

MicroStrategy Health Center can help you prevent, diagnose and fix problems in your MicroStrategy system. It detects known problems and provides an immediate solution to many of them. Health Center can email a system administrator if it encounters a problem with the system. In cases where Health Center cannot fix a problem immediately, it enables you to bundle relevant system information, such as log files, into a diagnostic package and transmit the package to MicroStrategy Technical Support for review and troubleshooting.

As part of a MicroStrategy installation, you can designate this machine as a Health Agent. After you complete the MicroStrategy installation, you can further define this machine as a Master Health Agent through the use of the MicroStrategy Configuration Wizard.

To define the machine as a Health Agent, provide the following information:

- **Port**: Type the port number to use to connect to the Health Agent machine. The default port is 44440.
- **Access Code**: Type the access code that must be provided by Health Center to access this Health Agent. If you leave this field blank, no access code is required to access this Health Agent.
- **Do not set the service account**: Select this check box to use the local system account as the Health Center service account. If you clear the check box to set a different Health Center service account, enter the following information:
- **Login**: A Windows login of the form Domain\User with full administrative privileges under which to run the Health Center service.

  The user account used to run Health Center 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.

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

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

**MicroStrategy Health Center Configuration**

You see this page only if you choose to install MicroStrategy Operations Manager. Specify the following configuration information to define the machine as a the Master Health Agent, which is responsible for most of the Health Center operations, such as scheduling system checks and transmitting diagnostics packages to MicroStrategy Technical Support:

- **Repository Path**: Click ... (the Browse button) to navigate to the location to store the Health Center repository. The repository contains configuration information about the Health Center system, such as the list of machines on the network and the MicroStrategy products they have installed, and also the destination for all exported diagnostics packages.

- **Customer Experience Improvement Program**: You can choose to enroll the installation in the Customer Experience Improvement Program:

  - **Join program**: Select this option to enroll the installation in the Customer Experience Improvement Program. Once enrolled, Health Center transmits anonymous data about your system to MicroStrategy. No report data or prompt answers are collected or transmitted. All information sent to MicroStrategy as a result of this program is stored in the Census subfolder of the Health Center Repository.

  - **I do not want to participate right now**: Select this option to opt out of the Customer Experience Improvement Program.

**Server Activation**

If you install 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
- MicroStrategy Narrowcast Server Delivery Engine
Welcome

Read the information on the Welcome screen and click **Next** to proceed to the next step.

Server Information

Specify information about your Intelligence Server installation. Enter the following characteristics:

- **Name**: Distinguishes the name of this Intelligence Server installation from any other Intelligence Server installations in your company
- **Location**: Physical location of the machine on which Intelligence Server is installed
- **Use**: Description of how Intelligence 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.
- For descriptions of what information to include in the other text fields, press **F1** to view the MicroStrategy online help.

Note the following:

- 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 licensed to use 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. For descriptions of what information to include in the text fields, press **F1** to view the MicroStrategy online help.
Note the following:

- Select the check box at the bottom of the page if you want 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 specified in the Installer Information and Contact Information pages. This email is sent upon completion of the installation process.

- 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 121** in **Chapter 4, 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 products stop functioning until you activate your installation. If you wait to activate your installation, you receive periodic reminders.

Once your installation is complete and you request an Activation Code, an email is sent to the email addresses you specified 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 123** in **Chapter 4, Activating Your Installation**.

When the Activation Code request process is finished, you are prompted to either view the **MicroStrategy Readme** or go directly to the MicroStrategy Installation Wizard Complete page. Click **Yes** to read the **MicroStrategy Readme** or **No** to go to the MicroStrategy Installation Wizard Complete page.

**CPU License Information**

You see this page only if both of the following statements are true:

- You are installing MicroStrategy Intelligence Server on a multi-processor machine.
- Your license is based on CPU and allows for more than one CPU.

Specify the number of CPUs that Intelligence Server is licensed to use.
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.

Note the following:

- 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, Mobile Server, and Operations Manager, page 190*. For additional configurations required to deploy Mobile Server, see the MicroStrategy Mobile Design and Administration Guide.

Note the following:
- 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 Operations Manager (ASP.NET) Setting**

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

Specify the Internet Information Services (IIS) virtual directory to be created for MicroStrategy Operations Manager. The default is MicroStrategyOM. The virtual directory is part of the URL used to access Operation Manager's administration console.

Operations Manager can be deployed using the same techniques used to deploy MicroStrategy Web, as described in *Deploying MicroStrategy Web, Mobile Server, and Operations Manager, page 190*.

Note the following:
- If you have a previous version of MicroStrategy Operations Manager installed on the machine, the new one that you install will use the same virtual directory that 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 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 Web Services Setting**

You see this page only if you choose to install MicroStrategy Web Services, which is required to run MicroStrategy Office, and only if you do not have a previous version of Web Services installed.

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

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

For information about deploying MicroStrategy Web Services ASP.NET and J2EE, see the MicroStrategy Office User Guide.

**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.

Refer to the MicroStrategy Narrowcast Server Installation and Configuration Guide for additional details about this setting.

**MicroStrategy Office URL Setting**

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

Specify the URL for MicroStrategy Web Services. The MicroStrategy Office client requires the MicroStrategy Web Services URL to access MicroStrategy projects. The URL depends on the name of the IIS virtual directory that you specified on the MicroStrategy Web Services page. To review the step in which the MicroStrategy Web Services page was specified, see MicroStrategy Web Services Setting, page 66.

Assuming that you kept the default value on the MicroStrategy Web Services page and you are installing on the same Web server machine that is hosting MicroStrategy Web, you should use the default URL provided:

http://localhost/MicroStrategyWS/MSTRWS.asmx
**MicroStrategy Office Configuration**

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

Select the check boxes to enable MicroStrategy Office for the associated Microsoft applications. You can configure MicroStrategy Office to integrate with Microsoft Excel, PowerPoint, and Word. The MicroStrategy Office toolbar is added to the Microsoft Office applications that you select.

**Start Copying Files**

This page displays the following information about your installation:

- Products that will be installed or updated
- Target directories in which the products are installed
- Name of the Windows Start menu program folder
- Virtual directories for MicroStrategy Web (ASP.NET), Narrowcast Server Subscription Portal, and Web Services
- URL for MicroStrategy Web Services
- Service accounts for MicroStrategy Narrowcast Server and Intelligence Server
- Location of the installation log file
- License details

Click **Install** to continue with the installation process, which can take several minutes depending on your computer's hardware configuration.

Click **Print** to print a copy of this information for your records.

**MicroStrategy Installation Wizard Complete**

If the option to restart your machine appears, select **Yes I want to restart my computer now** to ensure that the installation process finishes correctly. This is the recommended procedure, but you can also choose to continue without restarting.

Click **Finish** to complete the installation.

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

**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 55.*

**Performing a MicroStrategy Express installation**

The Express option installs MicroStrategy Secure Enterprise Platform with all the features of Analytics, Mobility, and Usher Security. 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 Usher Security configuration.

After installing the complete Platform, you will have MicroStrategy Analytics, Mobility, and Usher Security 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.
    - You have Windows administrative privileges.
    - 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.
  - The Usher Security app successfully installed on your iPhone or Android phone. Download the app from the App Store for your iPhone or Google Play for your Android.

- **MicroStrategy Secure Enterprise software requirements:**
  - 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 are not able to retrieve your badges from the Usher Security mobile app, or download your dashboards 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 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>

- **Usher Security 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 Usher Security:
  - An SSL Certificate Authority Root certificate and Intermediate certificates, appended into a (.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 Usher Security web services server.
  - Certificate paths provided during installation must not be changed since the Usher server depends on the certificates to operate.
  - To allow the Usher 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 the MicroStrategy.exe. Alternatively, you can locate and run the 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's Usher Security badge:
   - **First Name**: The first name of the Usher Security administrator.
   - **Last Name**: The last name of the Usher Security administrator.
   - **Email address**: The email address of the Usher Security administrator. This is the email address where you will receive the badge invitation for your Usher Security 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 Usher, provide the following information. If you do not have this information, contact your IT Administrator.
   - **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 Usher Security network.
Installation and Configuration Guide

- **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 Usher Security later, click **Skip** and refer to the Usher Security Documentation. The subsequent steps assume you have entered all the information required to set up Usher Security.

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 at [http://www.gnu.org/licenses/old-licenses/gpl-2.0.html](http://www.gnu.org/licenses/old-licenses/gpl-2.0.html).

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

10 Click **Install** to begin installing MicroStrategy Secure Enterprise. Installation takes approximately 20 minutes with the recommended hardware.

11 Once the platform has been installed, a restart is required to configure the new software and services. It is recommended that you restart your machine now by clicking **Finish**. The MicroStrategy software is not functional until after a restart.

12 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 Usher Security badges during this time period.

13 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 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. For steps, see **Activating Your Installation**.
By default, your machine’s timezone is change to Coordinated Universal Time (UTC).

To log in to Network Manager with your Usher Security app

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

2. From your mobile device:
   a. Download the Usher Security application for your iOS or Android device.
   b. Opening one email at a time, click each email’s invitation link to download each badge for your Usher Security network. Be sure that your mobile device is connected to the same network as your SMTP server.

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

4. Tap the QR scanner in your Usher app, and then scan the QR code. When prompted, select your Usher Network Manager badge.

5. From Network 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 your Usher Security mobile app. To scan the QR code, tap the QR scanner and scan the code.

   - If you did not configure Usher Security, you must log in using the MicroStrategy Administrator credentials, as described below:
     a. Click **Credentials** below the QR code.
     b. Type Administrator for the username and keep the password empty.
     c. 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 dashboards.
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 Express Installer randomly generated 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. 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 Narrowcast Server
- MicroStrategy Tutorial
- MicroStrategy Developer
- MicroStrategy Administration Tools

**Usher Security**

- Usher Server
- Usher Gateway Server
- Usher Network Manager
- Usher Analytics
- Usher Professional
Third-party software libraries

The MicroStrategy Express installation method installs a number of third-party software libraries in order to provide a streamlined user experience.

In addition to the MicroStrategy platform, the following software libraries are installed:

- Apache HTTP Server 2.4.18
- Apache Kafka 2.11-0.9.0.1
- Apache Tomcat 8.0.30
- Java SE Runtime Environment 8u77
- Logback 1.1.3
- Memcached 1.4.4
- MongoDB 3.2.9
- MongoDB drivers 8.0.1.21
- MySQL 5.6.28
- MySQL Connector/Python 2.1.3
- MySQL JDBC 5.1.22
- MySQL ODBC 5.3.4
- Node.js 4.5.0
- OpenSSL 1.0.2e
- PHP 7.0.2
- Python 2.7.11
- Redis 3.2.4
- Microsoft Visual C++ Redistributable 2010
- Microsoft Visual C++ Redistributable 2013
- Microsoft Visual C++ Redistributable 2015 Update 3
- XOM 1.2.10

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
- Network Manager: 443
- Usher Server 1-way: 1443
- Usher Server 2-way: 2443
- Gateway Server: 9501
- MySQL: 3306
- Intelligence Server: 34952

**Certificate information**

In addition to the certificates provided during installation, an Usher Signing CA is generated to support all PKI signing operations of the Usher 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 Usher 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\Usher Server\usherApps\shardIDM\conf\server.xml
  - SSLCertificateFile=HTTPSServerCertificate.crt
  - SSLCertificateKeyFile=HTTPSServerCertificate.key
  - SSLCACertificateFile=UsherTrustedCABundle.pem
- C:\Program Files (x86)\MicroStrategy\Usher\Usher Server\usherApps\shardGateway\conf\server.xml
  - SSLCertificateFile=HTTPSServerCertificate.crt
  - SSLCertificateKeyFile=HTTPSServerCertificate.key
  - SSLCACertificateFile=UsherTrustedCABundle.pem
- C:\Program Files (x86)\Common Files\MicroStrategy\Apache\Apache24\conf\extra\manager.usher.com.conf
  - SSLCertificateKeyFile HTTPSServerCertificate.key
  - SSLCertificateFile HTTPSServerCertificate.crt
- C:\Program Files (x86)\Common Files\MicroStrategy\php\php.ini
  - openssl.cafile=UsherTrustedCABundle.pem

In addition,
• Certificates from HTTPSServerCertificateRootCAChain.pem are added to Java Key Store

• UsherSAML.crt is saved to C:\Program Files (x86)\MicroStrategy\Usher\Usher 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 49* in Chapter 1, Planning Your Installation, for an installation and configuration checklist.

• The next chapter 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 4, 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 5, Configuring and Connecting Intelligence Server.
INSTALLING MICROSTRATEGY ON UNIX AND LINUX

This chapter describes the procedure for installing MicroStrategy on UNIX and Linux platforms. The installation procedure described in this chapter refer to installing MicroStrategy on Solaris, AIX, HP-UX, RedHat, and SUSE. This chapter covers the following sections:

Before installing MicroStrategy products, you should refer to Chapter 1, Planning Your Installation for important pre-installation information.

If you are installing MicroStrategy on Windows, refer to Chapter 2, Installing MicroStrategy on Windows.

Additionally, Chapter 10, Deploying OEM Applications explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.

For supporting installation information, refer to Installing MicroStrategy on Windows.

Installation procedures on UNIX and Linux

This section describes installing MicroStrategy on UNIX and Linux. The MicroStrategy products that you can install on UNIX and 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
• Usher Security Services
  ▫ Usher Security Server
  ▫ Usher Network Manager
• Usher Analytics
• Usher Professional

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 https://resource.microstrategy.com/msdz/default.asp. You can also access the MicroStrategy Developer Library from the MicroStrategy support site.

For more information about these products, see MicroStrategy products and components, page 1.

It is recommended that you install MicroStrategy products as the root user.
Be aware of the following:

- 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 register MicroStrategy Intelligence Server as a service. If the Intelligence Server is registered as an application during installation, the root user can register the server as a service after running the installation. For more information on running Intelligence Server as a service, see the System Administration Guide.

- Only a user with root permissions can install Usher 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 Choose Destination Location 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.

**Prerequisites**

To install MicroStrategy Usher, complete the prerequisites detailed in *Usher installation prerequisites*.

- If you are installing Usher Security Network Manager, you must create a database to manage Usher identities.

- If you are installing Usher Analytics, you must create a database to store Usher activity data.
Different methods of installation

MicroStrategy products can be installed on UNIX and 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 GUI mode

The GUI (graphical user interface) mode presents a user interface for each page in the MicroStrategy Installation Wizard. The following navigational buttons are displayed:

• **Next**: Proceed to the next page
• **Back**: Return to the previous page
• **Cancel**: Cancel the installation and close the MicroStrategy Installation Wizard
• **Finish**: Complete the setup and close the wizard

Using silent mode

You can perform a fully automated and unattended installation within the MicroStrategy platform when you do not have access to a UNIX or 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 UNIX or Linux environment, see *Chapter 9, Automated Installation on UNIX and Linux*.

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]. Press 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 press ENTER to proceed to the next page.
• Press 2 and then press ENTER to return to the previous page.
• Press 3 and then press 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 press ENTER to complete the setup and close the wizard.

**Installing with the MicroStrategy Installation Wizard**

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

<table>
<thead>
<tr>
<th>Note the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• You need root access permissions for installation if you have purchased the CPU-based MicroStrategy license.</td>
</tr>
<tr>
<td>• If you want to enable additional error and troubleshooting issue logging for the MicroStrategy installation routine, contact MicroStrategy Technical Support.</td>
</tr>
</tbody>
</table>

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

**To access the MicroStrategy Installation Wizard**

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

2. Browse to the MicroStrategy Installation folder. Depending on your UNIX or Linux environment, browse to one of the following folders:
   - **Solaris**: QueryReportingAnalysis_SunOS
   - **AIX**: QueryReportingAnalysis_AIX
   - **HP-UX**: QueryReportingAnalysis_HP-UX
   - **Linux**: QueryReportingAnalysis_Linux
Note the following:

- 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, Solaris, AIX, HP-UX, 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 46. Details on using a options.txt file to provide the location of the installation files are provided in Chapter 9, Automated Installation on UNIX and Linux and the parameters used to specify the location of the required installation files are described in Providing installation files for smaller installations, page 339.

3 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 321.

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

Note the following:

- 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 press ENTER, to proceed to the next page of the installation routine.

- At any time during the setup, you can click Cancel, or press 3 and then press ENTER, to quit the installation.
Language Setup

Specify the language to be used for the MicroStrategy installation and proceed to the next step.

Welcome

Read the information on the welcome screen and proceed to the next step.

MicroStrategy Installation Selection

This page is displayed only if there are installations of MicroStrategy software already present 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 UNIX and Linux machines. Additionally, you can also modify, repair, and upgrade existing MicroStrategy installations. This page 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 pages are the same as for a first-time installation, although some pages 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 hotfix**: Select this option to uninstall a MicroStrategy hotfix installation.

## License Agreement

Read the license agreement, and select to accept or decline the agreement. If you choose to 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 at https://software.microstrategy.com, contact your MicroStrategy Representative or contact MicroStrategy Technical Support at support@microstrategy.com.

## 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 $HOME/MicroStrategy if you do not have write access to /var/opt/MicroStrategy.
  - Do not install the MicroStrategy configuration files directly to your UNIX or Linux Home Directory ($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.

**Select Components**

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 Usher Analytics or Usher Mobile, components of MicroStrategy Analytics Enterprise are also installed. To install Usher Mobile, you must also install Usher Analytics.

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 1 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.

**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>mstr15.tzp</td>
<td>MicroStrategy Usher Server</td>
</tr>
<tr>
<td>Installation File</td>
<td>MicroStrategy Components That Require The Installation File</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>mstr16.tzp</td>
<td>MicroStrategy Usher Network Manager</td>
</tr>
<tr>
<td>mstr17.tzp</td>
<td>MicroStrategy Usher Analytics</td>
</tr>
<tr>
<td>mstr18.tzp</td>
<td>MicroStrategy Usher Mobile</td>
</tr>
</tbody>
</table>

- 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 9, Automated Installation on UNIX and Linux and the parameters used to specify the location of the required installation files are described in Providing installation files for smaller installations, page 339.

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.

**Missing Requirements**

This page is displayed only 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 on this setting and MicroStrategy’s recommendation on its value, see Supporting Intelligence Server memory allocation on Linux, page 32.

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 is displayed, some system resource limits are not configured to fully support the use of shared memory resources. To support this configuration, cancel the installation and refer to the limit recommendations provided in Configuring shared memory resources, page 31.

**System Requirements**

This page is displayed only 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 31.
**MicroStrategy Health Center Configuration**

MicroStrategy Health Center can help you prevent, diagnose and fix problems in your MicroStrategy system. It detects known problems and provides an immediate solution to many of them. Health Center can email a system administrator if it encounters a problem with the system. In cases where Health Center cannot fix a problem immediately, it enables you to bundle relevant system information, such as log files, into a diagnostic package and transmit the package to MicroStrategy Technical Support for review and troubleshooting.

As part of a MicroStrategy installation, you can designate this machine as a Health Agent. After you complete the MicroStrategy installation, you can further define this machine as a Master Health Agent through the use of the MicroStrategy Configuration Wizard. For information on configuring and using Health Center, refer to the System Administration Guide.

To define the machine as a Health Agent, provide the following information:

- **Port**: Type the port number to use to connect to the Master Health Agent machine. The default port is 44440.

- **Access Code**: Type the access code that must be provided by Health Center to access this Health Agent. If you leave this field blank, no access code is required to access this Health Agent.

- **UNIX Daemon**: Select this check box to configure this Health Agent as a daemon, so that the Health Agent process is constantly running in the background. This requires you to configure the Health Agent using an account that has root access privileges to the machine. If you do not have root access to the machine, clear this check box. This configures the Health Agent as an application. In this case, be careful not to stop the Health Agent process, so that the machine can remain part of the Health Center system at all times.

**MicroStrategy Health Center Master Health Agent Configuration**

You see this page only if you choose to install MicroStrategy Operations Manager. Specify the following configuration information to define the machine as a the Master Health Agent, which is responsible for most of the Health Center operations, such as scheduling system checks and transmitting diagnostics packages to MicroStrategy Technical Support:

- **Repository Path**: Click ... (the Browse button) to navigate to the location to store the Health Center repository. The repository contains configuration information about the Health Center system, such as the list of machines on the network and the MicroStrategy products they have installed, and also the destination for all exported diagnostics packages.

- **Customer Experience Improvement Program**: You can choose to enroll the installation in the Customer Experience Improvement Program:
  - **Join program**: Select this option to enroll the installation in the Customer Experience Improvement Program. Once enrolled, Health Center transmits
anonymous data about your system to MicroStrategy. No report data or prompt answers are collected or transmitted. All information sent to MicroStrategy as a result of this program is stored in the Census subfolder of the Health Center Repository.

- **I do not want to participate right now:** Select this option to opt out of the Customer Experience Improvement Program.

**CPU License Information**

This page is displayed only if the Intelligence Server license has a CPU number limitation.

Specify the number of CPUs that Intelligence Server is licensed to use.

**Software 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

**Welcome**

Read the information on the welcome screen and proceed to the next step.

**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.

Note the following:

- 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.

Note the following:

- 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 121 in Chapter 4, 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 123 in Chapter 4, Activating Your Installation.

**Usher Security Server Settings: Step 1**

You see the Usher Security Server Settings page if you are installing Usher Security Server. Usher Security Server installs a database which is a system of record for individual Usher identities. Use this page to provide the configuration parameters for the Usher Security 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. The installation of Usher Server includes a ROOT.war file. You can use this .war file to deploy your Usher Server on your Tomcat web application server.

Define the database connection information for the database that will store the Usher Security 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 Usher Security Server database.
- **Log Database Instance**: The name of the database that stores log information for the Usher Security Server.

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.

**Usher Security Server Settings: Step 2 for ports and certificates**

You see the Usher Security Server Settings page if you are installing Usher Security Server. This page allows you to set up a trust relationship for Usher Security Server using the Public Key Infrastructure (PKI).

Define Usher Security Server’s HTTPS port using the following settings:

• **Server (one-way SSL) authentication only**: The default port is 443.

• **Client and Server (two-way SSL) mutual authentication**: The default port is 1443.

Provide the location of Usher Security Server’s certificate files using the following settings:

• **Public Key SSL Certificate File**: The SSL 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. By default, the path is `USRER_SERVER_INSTALL_PATH/usherApps/shardIDM/conf/Server_ca.crt`.

• **Private Key File**: The 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. By default, the path is `USRER_SERVER_INSTALL_PATH/usherApps/shardIDM/conf/Server_ca.key`.

• **SSL Certificate Chain**: The SSL certificate chain is the list of Secure Sockets Layer (SSL) certificates, from the root certificate to the end user certificate, issued by a certificate authority (CA). The file contains the information in a pem format. By default, the path is `USRER_SERVER_INSTALL_PATH/usherApps/shardIDM/conf/Server_ca.pem`.

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

**Usher Security Server Settings: Step 3 for Gateways**

You see the Usher Security Server Settings page if you are installing Usher Security 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 Usher Security Server that is used for synchronizing your Usher users from Microsoft Active Directory. The Agent Gateway:

• Establishes a trusted relationship between Usher Security Server and the Usher Agent for Microsoft Active Directory, which is the application that communicates between your Active Directory server and Usher Security Server.

• Manages communication between the Usher Agent and Usher Security Server.
Communicates the Usher Agent’s status to Usher Security Server.

Provide the **Agent Gateway (one-way SSL) Authentication Only Port** (the default is 9501), and then provide the location of the Agent Gateway’s certificate files using the following settings:

- **Gateway SSL Certificate File:** The SSL certificate is used to encrypt the data and enforce authentication. The file contains the certification part of the signed certificate in a crt format. By default, the path is `{PATH}/usherApps/shardGateway/conf/Gateway_ca.crt`.

- **Gateway SSL Key File:** The SSL 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. By default, the path is `{PATH}/usherApps/shardGateway/conf/Gateway_ca.key`.

- **Gateway SSL Certificate Chain:** The SSL certificate chain is the list of Secure Sockets Layer (SSL) certificates, from the root certificate to the end user certificate, issued by a certificate authority (CA). The file contains the information in a pem format. By default, the path is `{PATH}/usherApps/shardGateway/conf/Gateway_ca.pem`.

The `{PATH}/usherApps/shardGateway/conf/Gateway_ca.crt` is the path specified for the install directory.

**Usher Network Manager Settings**

You see the Usher Network Manager Settings page if you are installing Usher Network Manager.

Specify the location of the **Apache Directory**. The Installation 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.

Usher Network Manager installs a database to manage Usher identities. You can choose to use the same database connection as Usher Security 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 Usher Network 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.

To test the database connection, click the **Test** button.
By default, Usher Network Manager is configured as an HTTP connection. For a more secure connection, it is strongly recommended that you configure it as an HTTPS connection. For steps, see your third-party Apache documentation.

**Usher Analytics Settings**

You see the Usher Analytics Settings page if you are installing Usher Analytics. Usher Analytics installs a database to store Usher activity data. Use this page to provide the configuration parameters for the Usher Security Server to communicate with the Usher Analytics database.

You can choose to use the same database connection as Usher Security 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 Usher Analytics machine.
- **User Name**: The account name for the database user that administers the database.
- **Password**: The password for the database user specified above.

The Usher Analytics database needs to be on the same MySQL instance as the Usher Security Server database.

To test the database connection, click the **Test** button.

**Start Installer Operation**

This page 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.

**MicroStrategy Install Wizard Complete**

When the MicroStrategy installation has completed, you can select the following:

- **Run Usher Configuration.** Available if Usher Network Manager was installed.
- **View the MicroStrategy Readme** for the latest updates.
- **Run MicroStrategy Configuration Wizard**, which allows you to configure your MicroStrategy production environment. For more information, see *Chapter 5, Configuring and Connecting Intelligence Server*.

Click **Finish** to complete the installation.
Unique post-installation configurations

MicroStrategy supports many different UNIX and 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.

- *Migrating Intelligence Server from Windows to UNIX or Linux, page 97*
- *Create links for Intelligence Server startup in SUSE Linux, page 97*
- *Supporting fonts for documents, exported reports, and graphs, page 98*
- *Starting PDF Exporter Service*

Migrating Intelligence Server from Windows to UNIX or Linux

If you are installing MicroStrategy Intelligence Server on UNIX or Linux and previously had Intelligence Server installed on a Windows platform, it is strongly recommended you modify certain system tuning settings. These memory and cache settings govern and can optimize the performance of Intelligence Server and MicroStrategy projects in your 64-bit UNIX or Linux environment. For more information on these system tuning steps, see the *After the Upgrade* chapter of the [*Upgrade Guide*](#).

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:
   ```bash
   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:
   ```bash
   ln libcrypto.so.0.9.7 libcrypto.so.4
   ```
Supporting fonts for documents, exported reports, and graphs

When Intelligence Server is running on a UNIX or 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 UNIX and 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 263.

Starting 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:

- 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.

For other configuration settings, see Export Engine configuration.

Verifying installation

During the installation routine, 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 due to which the errors occurred.

**Directory structure**

The following table describes 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 is complete.</td>
</tr>
<tr>
<td>INSTALL_PATH/CommandManager</td>
<td>MicroStrategy Command Manager files. This is the default directory for Command Manager but another location can be selected during installation.</td>
</tr>
<tr>
<td>INSTALL_PATH/Help</td>
<td>Documentation and Help for MicroStrategy products.</td>
</tr>
<tr>
<td>INSTALL_PATH/GISConnectors</td>
<td>MicroStrategy Portlet files. This is the default directory for the Portlets but another location can be selected during installation.</td>
</tr>
<tr>
<td>INSTALL_PATH/IntelligenceServer/bin</td>
<td>Intelligence Server-specific binary files.</td>
</tr>
<tr>
<td>INSTALL_PATH/Mobile</td>
<td>MicroStrategy Mobile and Mobile Server JSP files. This is the default directory for Mobile but another location can be selected during installation.</td>
</tr>
<tr>
<td>INSTALL_PATH/PDFGeneratorFiles</td>
<td>Support files (fonts) for the PDF generation feature of Intelligence Server.</td>
</tr>
<tr>
<td>INSTALL_PATH/Portlets</td>
<td>MicroStrategy Portlet files. This is the default directory for the Portlets but another location can be selected during installation.</td>
</tr>
<tr>
<td>INSTALL_PATH/SystemManager</td>
<td>MicroStrategy System Manager files. This is the default directory for System Manager but another location can be selected during installation.</td>
</tr>
<tr>
<td>PATH/Directory</td>
<td>Contents</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INSTALL_PATH/Usher</td>
<td>MicroStrategy Usher files.</td>
</tr>
<tr>
<td>INSTALL_PATH/WebServicesJ2EE</td>
<td>MicroStrategy Web Services deployment path.</td>
</tr>
<tr>
<td>INSTALL_PATH/WebUniversal</td>
<td>MicroStrategy Web deployment path.</td>
</tr>
<tr>
<td>INSTALL_PATH/_jvm</td>
<td>The Java Runtime Environment (JRE) to be used by the Java applications. It provides the requirements for executing a Java application, a Java Virtual Machine, core classes, and supporting files.</td>
</tr>
<tr>
<td>INSTALL_PATH/_uninst</td>
<td>Launch files for uninstalling MicroStrategy.</td>
</tr>
<tr>
<td>INSTALL_PATH/bin</td>
<td>64-bit binary files.</td>
</tr>
<tr>
<td>INSTALL_PATH/bin32</td>
<td>32-bit binary files.</td>
</tr>
<tr>
<td>INSTALL_PATH/help</td>
<td>Online help files.</td>
</tr>
<tr>
<td>INSTALL_PATH/jar</td>
<td>Java libraries.</td>
</tr>
<tr>
<td>INSTALL_PATH/lib</td>
<td>64-bit binary libraries.</td>
</tr>
<tr>
<td>INSTALL_PATH/lib32</td>
<td>32-bit binary libraries.</td>
</tr>
<tr>
<td>INSTALL_PATH/locale</td>
<td>ODBC support messages.</td>
</tr>
<tr>
<td>LOG_PATH</td>
<td>MicroStrategy application log files, which includes Intelligence Server log files.</td>
</tr>
</tbody>
</table>

**MicroStrategy Messaging Services**

Messaging Services is a component that is coupled with the Intelligence Server during installations and upgrades for Linux environments. For new installations, Messaging Services is configured out-of-the-box and runs automatically after the installation is completed. If you are upgrading to the 10.6 Feature release, Messaging Services requires post-upgrade configuration for the Zookeeper server, Kafka server, and Consumer services.

All logging happens through Messaging Services and is now placed in the log/DSSError file.

For steps on installing and configuring Messaging Services, see the following steps.

**Prerequisites**

- For new installations, you must be installing on a Unix/Linux environment.
- For upgrades, you must have a Unix/Linux environment.
- To configure Messaging Services, Command Manager is needed.
Installing Messaging Services as part of a new installation

1 To install Messaging Services, check the box labeled MicroStrategy Intelligence Server on the Select Components page. By default, Messaging Services is installed whenever Intelligence Server is selected for installation.

2 Complete the installation. For instructions about installing on Unix /Linux environments, see the Installation and Configuration Guide.

3 Once the your installation is complete, Messaging Services is running by default and the Intelligence Server is logging through Messaging Services.

To use Intelligence Server for logging

You can elect to disable Messaging Services and use the Intelligence Server for logging. To disable, refer to the Command Manager Script table for the correct script to stop MicroStrategy Messaging Services.

If you want to use Messaging Services at any point in time after disabling it, you need to restart the Messaging Services to begin logging. Refer to the Command Manager Script table for more information.

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 start MicroStrategy Messaging Services. If not, the Intelligence Server continues to write logs to the original log file. For information about the correct Command Manager script, refer to the MicroStrategy Messaging Services Script table.

Command Manager Scripts for MicroStrategy Messaging Services

<table>
<thead>
<tr>
<th>To start Messaging Services</th>
<th>To stop Messaging Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER SERVER CONFIGURATION ENABLEMESSAGINGSERVICES TRUE</td>
<td>ALTER SERVER CONFIGURATION ENABLEMESSAGINGSERVICES FALSE</td>
</tr>
<tr>
<td>CONFIGUREMESSAGINGSERVICES</td>
<td></td>
</tr>
</tbody>
</table>

MicroStrategy Messaging Services 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.

1 Locate the consumer file on your Intelligence Server:

   <Installationpath>/MicroStrategy/_jre/bin/java -jar
   KafkaConsumer.jar

2 You need to change the following:
The log folder where you intend to store the log files
If you don't want to use the default Kafka server connection, you can input 'Y' and specify your information.
MicroStrategy Messaging Services machine IP address and server port

Usher Security Installation and Configuration

The steps below show you how to install and configure MicroStrategy Usher Security on Linux. You can install Usher Security as part of the MicroStrategy platform. If you chose to only install Usher Security, be aware that certain MicroStrategy components are required for Usher Analytics and Usher Professional. These are installed in the background. If you are upgrading your existing MicroStrategy Usher products, refer to the Upgrade Guide. For all Usher-related Help, see the Usher Online Help.

Overview of Usher installation and configuration process

The high-level steps to install and configure Usher are listed below:

1. Gather the information to create a certificate signing request. To install MicroStrategy Usher, you need to obtain a series of trusted CA-signed certificates. Contact your security administrator to complete this process. Refer to Installation considerations for more information.

2. Ensure that the prerequisites are complete. See Usher installation prerequisites.

3. Complete the pre-installation steps to configure your Linux system to allow MySQL and to grant privileges to each product. For steps, see:
   a. To configure Linux for Usher Security
   b. To set up Apache on your Linux machine
   c. To set up your timezone
   d. To set up your Fully Qualified Domain Name
   e. To open ports in your firewall
   f. To install memcache
   g. To download and set up Java Developer Kit
   h. To download and set up Apache Tomcat version
   i. To download and set up PHP
   j. To install and set up MySQL
   k. To grant privileges to your MicroStrategy products in MySQL
4 Install MicroStrategy Secure Enterprise Linux. For steps, see *To install Usher Security*.

5 Deploy the Usher Server and complete configuring Usher Server and Network Manager. For steps, see:
   a  *To deploy your Usher Server*
   b  *To enable https for Usher Network Manager*
   c  *To set up Usher Network Manager*
   d  *To set up your Usher Gateway Server*

**Usher installation prerequisites**

- You have previously obtained the MicroStrategy installation package.
- You have obtained either an universal installation key or an Usher installation key.
- You are using Red Hat Enterprise Linux 6, or a similar flavor, as your operating system. To find your Linux version, run the following command: `cat /etc/*-release`.
- You have Linux root user permission to complete the pre-installation steps.
- You have Apache HTTP 2.2.x on your Linux system.
- You need to install Oracle Java Developer 1.7.x.
- You need to install Tomcat 8.x and the supporting libraries.
- You need to install PHP 5.4.x and the supporting libraries.
- You need admin access to MySQL 5.7.x
- You can supply a Google key for Usher Analytics. This key is used to configure Google Geocoding in Usher Analytics.
- If you are configuring Usher Analytics and Usher Professional after installing Usher Security, you need to install Python 2.7.

**MicroStrategy Usher Security pre-installation steps**

- Do not save files in your `\root\` folder path during installation, because doing so will not allow you to successfully install Usher Security.

To successfully install Usher Security, your Linux systems need to be configured to allow PHP to connect to MySQL and grant privileges to each product.

---

**To configure Linux for Usher Security**

1 Disable SELinux by navigating to:
To set up Apache on your Linux machine

If you are using Red Hat Linux, Apache has already been installed as part of the system. If you are using another version of Linux, you need to install Apache.

You need Apache version 2.2.x. To check the version of Apache that you have installed, enter the following command: `> httpd -v`.

- If you don't have Apache already installed on your system, enter `yum install httpd` to install, then enter the following command to start the service:
  ```
  > service httpd start
  ```

  Ensure that you have installed version 2.2.x by checking the version number of Apache by entering: `> httpd -v`.

To set up your timezone

Set your system timezone to GMT `localtime` for use with the Usher Server.

1. Enter the following commands:
   ```
   > cd /etc
   > rm localtime
   > ln -s /usr/share/zoneinfo/GMT localtime
   ```

To set up your Fully Qualified Domain Name

The steps below require you to work with your IT department to change your Linux hostname. You are also required to 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 to learn about your company's policy on Fully Qualified Domain Names (FQDN).

2. Have your Linux hostname changed to a FQDN per your company's policy. Then have your new FQDN and IP address mapped.

3. Following the direction of your IT department, obtain a CA-signed certificate from a trusted vendor.
You may be asked for a certificate request that contains your company's information. If so, follow the steps below:

a  Create a certificate request (*.csr) and private key (*.key) and provide this to your vendor in the form requested.

b  Provide your company's information to your certificate vendor. This information is incorporated into your certificate request.

4 Upon receiving your certificate from the third-party vendor, create a *.pem by combining your certificate (*.crt) and private key (*.key). If you are installing on UNIX, enter the following command:

   > cat intermediate.crt server.crt server.key > server.pem

5 To complete a successful Usher Security installation, you need to create a self-signed certificate. To do this, generate a certificate (*.crt) and a private key (*.key). If you are installing on UNIX, enter the following command:

   > openssl req -new -newkey rsa:2048 -days 365 -nodes -x509 -keyout self.key -out self.crt

---

**To open ports in your firewall**

To allow Usher Security to connect with the Usher Server, you need to open ports in your system's firewall. Contact your IT department for specific instructions. It is recommended to use the default SSL port of 443.

---

**To install memcache**

You need memcache to install Usher Security. This allows the database tables that you create to be saved in the memory. After installing memcache, you need to configure the modules for Usher Security.

1 You can install memcache by using either **yum** or **rpm**. Select a method and enter the corresponding commands:

   - **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 memcache by changing the following in your memcache file:

```
vi /etc/sysconfig/memcached
PORT="11211"
USER="memcached"
MAXCONN="1024"
```
CACHESIZE="64"
OPTIONS=""

3 Restart memcache by entering: `service memcached start`.

4 Verify that the server is running by checking on the status.

To download and set up Java Developer Kit

MicroStrategy supports Oracle Java SE Development Kit 7. Other versions of Java Developer Kit (JDK) may not have required components for the MicroStrategy platform.

1 Check to see if you have a version of Java DK already installed. Run the following command:
   `update-alternatives --config java`

2 Create a directory on your local computer where you want to install the JDK.

3 Navigate to http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html, and select the correct 7u79 download version for your OS.

4 Click to download both the `tar.gz` file and `.rpm` file, and place them in your Java directory.

5 Click to start the installation process and follow the prompts.

6 Verify that the JDK was successfully installed. Enter the following commands:
   - `> which java`
   - `> java -version`.

To download and set up Apache Tomcat version

To verify whether Tomcat is already installed, run the following command: `echo $CATALINA_HOME`, which gives you the location of your Tomcat installation path.

Navigate to http://tomcat.apache.org/download-80.cgi, and under Core in Binary Distributions, click to download the `tar.gz` file.

Click to install and follow the installation prompts, noting the location of the download location for Tomcat. Next, you must install third-party libraries into your Tomcat directory, and note the location of the libraries. Open each of the following websites and follow the prompts for installation:

1 Go to http://www.xom.nu and click to download the complete tar.gz file. Install the file in your Tomcat library.
To download and set up PHP

You may encounter problems while setting up PHP on UNIX. For help troubleshooting error messages, refer to your IT department and third-party reference material for more information.

Prior to downloading the installation file, you must configure your system by installing prerequisites. These are required to successfully build PHP for MicroStrategy products.

First, use your subscription manager to enable SCL:

```
subscription-manager repos --enable rhel-server-rhscl-6-eus-rpms
```

Enter the following yum installation commands to install php 5.4 and modules:

```
yum install php54 php54-php php54-php-gd php54-php-mbstring
```

You must also install the updates database module for MySQL. To do this, enter:

```
yum install php54-php-mysqlnd
```

Next, disable the loading of php 2.3 Apache:

```
mv /etc/httpd/conf.d/php.conf /etc/httpd/conf.d/php.conf.old
```

Restart Apache to make the changes:

```
service httpd restart
```
5 After the installation is complete, you need to enable mod_rewrite. To do this, enter 
<code>/InstallPath/etc/httpd/conf/httpd.conf</code> to modify the file. The 
parameters are as follows:

```html
<!-- Directory " /var/www/html" -->
Options FollowSymLinks
AllowOverride All
</Directory>
```

6 Verify that the required PHP modules have been installed correctly.

a Enter the following commands:

1. `>php -v` to check the version of PHP.
2. `> php -m` to check the modules installed:
   - curl
   - zlib
   - openssl
   - gd
   - mbstring
   - mysql
   - pdo-mysql
   - zip
   - PDO
   - exit
3. `> php -r 'print_r(gd_info());'` to verify the GD version and 
   JPEG support.

b Add <code>/etc/httpd/conf/httpd.conf</code> by executing the following commands:

1. `LoadModule php5_module modules/libphp5.so`
2. `AddType application/x-httpd-php .php`

c Create <code>test.php</code> in <code>/var/www/html</code> by entering the following commands:

1. `<?php
2. `phpinfo();`
3. `?>`

d Restart the service to complete the changes.
In your local browser, navigate to `http://localhost/test.php`. You should see the PHP landing page, which signals that the verification is complete.

**To install and set up MySQL**

To successfully install Usher Security, you need to create a MySQL user and grant access to each MicroStrategy product. MicroStrategy supports MySQL Community Server version 5.6.23 and above. If you already have MySQL installed on your system, you still need to install 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, open `https://dev.mysql.com/downloads/mysql`. From the drop-down list, select the correct platform of MySQL Community Server for your system. Find your OS version, and click to download the RPM file.

2. Navigate to `http://dev.mysql.com/downloads/repo/yum/`, and click to 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 need to 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:$CLASSPATH`.
   - For C shell, enter: `> setenv CLASSPATH /home/suer/mysql-connector-java-5.1.34-bin.jar:$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` to install the `convert_tz` functions.
• If an error message displays that the function is unable to load, check to see 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 mysql start

6 Set the root password for MySQL.
   a Enter the following command: `mysql_secure_installation`.
   b Enter: `mysql -u root -p`, followed by a password of your choice. Be sure to note the password, as this 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.
   a Enter: `GRANT ALL`.
   b To create the mstr user, enter: `create user 'username'@'localhost' identified by 'username';`.
   c To grant the mstr user access, enter: `grant all on abcd to 'username'@'localhost';`, where `abcd` is the name of your local environment.

8 If you are using Usher Analytics to examine the data collected on your Usher Security server, you need to install crontab for MySQL.
   a Check that this directory exists: `mysql_tzinfo_to_sql` /usr/share/zoneinfo | /usr/bin/mysql -h127.0.0.1 -P3306 -umstr -p mysql.
   b Begin the installation by entering: `yum install crontab*`.
   c After the installation is complete, restart crontab by entering: `service crond start`.

---

To grant privileges to your MicroStrategy products in MySQL

You need to grant required privileges to the MySQL user. For each MicroStrategy product you plan to install, you need to grant access to make changes to the database. For example, if you are installing MicroStrategy Usher Security, MicroStrategy Web, and MicroStrategy Developer, then you need to grant MySQL access to each product.

1 In a command window, open MySQL.

2 Select a username to use, and type the following command:

   `create user YourUsername@localhost identified by YourUsername;`.

   Replace the italicized variables with the user name that you selected and your local environment name.
3 In MySQL workbench, on your local environment, click the management tab, then navigate to Users and Privileges.

4 Click to add an account.

5 In the login name field, type your user name.

6 Select standard as the method of authentication.

7 Type your password in the password field, and then confirm your password. (This is the password that you created in the previous steps.)

8 Click **Apply**.

9 Click **Administrative Roles** and select the applicable roles for the product(s) that you are installing.

10 Click to grant access to the required privileges. The privileges shown below must be granted for each MicroStrategy product you install:

   - ALTER
   - ALTER ROUTINE
   - ALTER VIEW
   - CREATE
   - CREATE TEMPORARY TABLES
   - CREATE VIEW
   - DELETE
   - DROP
   - EVENT
   - EXECUTE ROUTINE
   - GRANT OPTION
   - INDEX
   - INSERT
   - LOCK TABLES
   - SELECT
   - SHOW VIEW
   - SUPER
   - UPDATE

For example, if you are installing MicroStrategy Web and MicroStrategy Usher Security, each privilege must be manually granted to each of the products.
For Usher Security, if you want network admin users to be able to create new user accounts on your network, click CREATE USER for that role.

Installing MicroStrategy Usher Security

After completing all the pre-installation steps, you can begin the installation for MicroStrategy Secure Enterprise Linux.

To install Usher Security

2. Extract the downloaded file by running the following command:
   ```bash
tar -xvf filename.tar.gz
   ```
   Running this command extracts the file in the current directory.
3. Run the setup.sh file.
4. When prompted, select new installation. Click to accept the license agreement.
5. Enter your license key. If you are using a universal key, all MicroStrategy products available for installation are displayed. If you are using an Usher key, only the Usher Security products are installed. Click Next.
   When using an Usher key for installation, the MicroStrategy platform is installed in the background. These products are required for you to use Usher and Usher Analytics properly.
6. For the Home, Install, and Log directories, it is recommended to use the default directory locations. Click Next.
7. Select the MicroStrategy products to install, then click Next.
8. Click to browse to your Tomcat location. You can find the location of your Tomcat directory by running the following command:
   ```bash
echo $CATALINA_HOME
   ```
9. In the Database Connection fields, provide your MySQL connection information:
   - **Server**: Your server IP Address
   - **Port**: Your MySQL port (default port number is 3306).
   - **User**: Your MySQL username.
   - **Password**: Your password (if you set one to be required for MySQL user).
   - **Server DB Instance**: Leave the default value (usher_server).
   - **Log DB Instance**: Leave the default value (usher_server_log).
10 Click **Next** to test your connection. To continue, you must have a successful connection.

11 In the SSL ports, set the default for your one-way and two-way ports:
   - The default for a one-way port is 443.
   - The default for a two-way port is 1443.

Record the port numbers, as they are needed for future steps.

12 Click to upload your CA-signed security certificate files (.crt). These signed certificates were signed by an external certificate authority, like Verisign and Thawte.
   - **SSL Certificate file**: Your .crt file
   - **Private key file**: Your .key file
   - **SSL Certificate Chain file**: Your .pem file

   To ensure that the SSL Certificates are valid for your Usher installation, you can run a checksum on both the .crt and .key files and verify that the results are identical.

13 Click **Next**.

14 Note the value provided for Agent Gateway port, as this value is needed for future steps.

15 By default, the **Use the same SSL certificate as Usher Server** check box is selected. If you want to use a different set of signed security certificates, click to browse and select the new security certificates.

16 Click **Next**.

17 In the **Apache Directory** field, keep the default value.

18 Unless you previously modified your Apache user and user group, keep the default value.

19 For **Database Connection**, you can use the same connection as your Usher server by selecting the check box. To use a different database instead, provide the required MySQL connection information.

20 Click **Next**, which verifies the Apache directory and MySQL.

21 For Usher Analytics, you can use the same MySQL database as before by selecting the check box. To use a different database instead, provide the required MySQL information. Click Next.

22 For the **MicroStrategy Health Center Repository Path**, you must specify a path (for example, /opt/hc, to continue.

23 Click **Next** until you reach the installation summary page, then click to install.

24 After the installation page, select the check box to launch the Usher Configuration tool.
25 If you have installed other MicroStrategy products with a universal key, select the check box to run the MicroStrategy Configuration Wizard.

26 Click Finish.

27 After the installation process is completed, Usher Configuration tool is automatically launched. You must first deploy your Usher server before using the Configuration tool is receive the key for your MicroStrategy server.

**Usher post-installation steps**

After you have installed Usher Server and Network Manager, you need to deploy the Usher Server and complete the configuration of Usher Server and Network Manager.

**To deploy your Usher Server**

1 In a command window, navigate to the installation location of Usher Server:

   ```
   > cd /installationpath/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:

   `<serverpath>/Usher/UsherServer/usherApps/shardIDM/webapps`

   If Tomcat does not start, refer to the catalina.out log located at `<serverpath>/Usher/UsherServer/usherApps/shardIDM/logs` to view details of the problem.

3 Copy the logback classic 1.1.2 jar file from your Tomcat directory to your Usher server root. To do this, enter the following command:

   ```
   ```

4 Copy the CA-signed certification over to your Usher server root, and rename the certification to saml2.crt. To do this, enter the following one-line command:

   ```
   > cp /<serverpath>/<certificatename>/Usher/UsherServer/usherApps/shardIDM/webapps/files/saml2.crt
   ```

5 Restart your Usher server to make the changes permanent. To do this, enter the following commands:
• > cd
    /<serverpath>/Usher/UsherServer/usherApps/shardIDM/bin
• > ./tomcat.sh restart

6 Check to see that your Usher server has restarted. In a web browser, type the following:

https://FQDN:1-way port

where

• FQDN is your domain name
• 1-way port is your SSL one-way port number (443 is the default one-way port number)

An example is https://yourdomainname:443.

7 Once your Usher server has been deployed, navigate back to your Usher Configuration tool. Fill out the required fields with your company information to receive the activation key for your MicroStrategy server.

To enable https for Usher Network Manager

Enabling https for Usher Network Manager is optional. If you elect to enable https, then you must also enable SELinux.

To enable SELinux

1 Open the SELinux config file:
2 vi /etc/selinux/config

Replace disabled
3 SELINUX=disabled -> SELINUX=enforcing,
4 Restart SELinux to make the changes.

To enable https

1 Install ssl_module by entering the following command:
   > yum install mod_ssl
2 Check to see that the module was enabled during the installation:
   > httpd -M | grep ssl_module
3 For Network Manager to communicate with your Tomcat port, you need to change the ssl port. To do this, enter the following command:
   > vim /etc/httpd/conf.d/ssl.conf
4 Change your port numbers from ssl port to Tomcat. For example, listen 443 becomes listen 9443.

5 The Network Manager configuration file needs to be edited in order to wrap with the virtual host. Enter the edit command:

   > vim /etc/httpd/conf.d/manager.usher.com.conf

6 Refer to the following parameters to determine what values to change:

   <VirtualHost *:9443>
   SSLEngine on
   SSLCertificateKeyFile /opt/usher/ssl/server.key #change this
   SSLCertificateFile /opt/usher/ssl/self.crt #change this
   SSLCAChainCertificateFile /opt/usher/ssl/server.crt #change this
   SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
   Alias /networkmanager /opt/MicroStrategy/Usher/UsherNetworkManager/networkmanager #change this
   <Directory /opt/MicroStrategy/Usher/UsherNetworkManager/networkmanager>
   #change this
   AllowOverride All
   </Directory>
   </VirtualHost>

To set up Usher Network Manager

Prerequisites

You need the following to complete the Usher configuration:

- The SSL port numbers from your MicroStrategy installation steps.
- Your self-signed certificate.
- The key for your self-signed certificate.
- Your trusted vendor CA-signed certificate.
- You have a SMTP server
- If you plan to use Usher Network Manager with Google Maps, your Google Maps API key.
• The Usher Security application for your smartphone has been downloaded.

1 Open your Usher Network Manager page by typing the following in a browser window:

http://yourFQDN:443/networkmanager/managesystem

If you have enabled https, your URL should be as follows:

https://yourFQDN:443/networkmanager/managesystem

2 Enter your one-way port server URL, making sure you use your FQDN (for example, https://FQDN:443), 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 Usher), and click Log In.

4 Fill out the following fields with your company information:
   • Security Server Host: Your auto-populated FQDN and one-way port number. This value cannot be edited.
   • System Name: Your system name.
   • Usher File Directory: Keep the default value.
   • SSL Certificate Authority Certificate: Your trusted vendor CA-signed .crt.
   • SSL Certificate Authority Key: The key for the trusted vendor CA-signed certificate.
   • 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 Usher Server installation. Keep the default value.
   • SAML Certificate: Your self-signed .crt certificate that was generated using Openssl.
   • SAML Key: Your self-signed *key that was generated using Openssl.
   • SMTP Server: Your company’s SMTP server.
   • 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.
   • Email Sender Address: The email address that you are using to send the badge invitations for your Usher network.
• **Security Server Host**: The address of your two-way SSL server (default port is 1443). For example, yourFQDN:1443.com. For the Security Server Host, https:// is not needed.

• **Gateway Host**: Keep the default URL and enter your Agent Gateway port value (default value is 9501). For example, yourFQDN:950.com For reference, see *To install Usher Security*. 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 Usher Network, enter the third-party key.

• **Memcache**: No node is required for the Usher configuration.

• **iPad Configuration Link**: If you have already set up Usher Professional, enter your iPad URL link.

• **iPhone Configuration Link**: If you have already set up Usher Professional, enter your iPhone URL link.

   If you do not set the SMTP server and port, the Usher server cannot send emails

5 Click Next.

6 Fill in the following fields:

   • **Network Manager Host**: Your FQDN URL. Keep the default value.
   
   • **Network Manager Path**: Keep the default value.
   
   • **Security Server Host**: Your security server address.
   
   • **Network Creation**: If you do not want your Usher users to create their own network on your Usher server, select the Require Authentication check box.
   
   • **MicroStrategy Managed Instance**: If your system is housed in Amazon Web Service, select the Restrict LDAP Configuration check box.
   
   • **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 Click to create an Usher Admin.
To upload a photo for your Admin account, click to select a file. Supported image formats are .png, .jpeg, and .jpg.

Enter your first and last names in the correct fields.

In the Email Address field, type the email address to associate with your Usher Admin account.

Click Create. Usher sends an email invitation to the email address that you provided.

Open the email in your smartphone, then click Get My Badge, which opens the Usher Security application.

In a web browser, navigate to http://yourFQDN/networkmanager, and scan the QR code displayed.

After you log in, you can create the security network and add users. See the Usher Online Help for detailed steps about how to finish setting up your Usher Network Manager. After you have successfully created an Usher Administration account, you must deploy the Gateway Server before creating new Security Networks.

To set up your Usher Gateway Server

After you have configured Usher Server and Network Manager, you can deploy the Gateway. These steps must be completed in Linux, not a Windows environment.

Prerequisite

Append your self-signed certificate to the CA-signed .pem file. This allows you to properly set up the Usher Gateway. For steps, see To set up your Fully Qualified Domain Name.

1 In a command window, navigate to the Usher Server Gateway using the following:
   
   cd
   /
   <installpath>/Usher/UsherServer/usherApps/shardGateway/bin

2 Start Tomcat:
   
   > ./tomcat.sh start

   This deploys the gateway.war file.

3 Check to confirm that the gateway was deployed successfully. In a browser window, type the following command:
   

4 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 Active Directory with your Usher Network, configure Usher Analytics, configure your MicroStrategy Mobile Server, and other Usher Network Manager help, refer to the Usher Help.

**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 UNIX and Linux, page 50* in *Chapter 1, Planning Your Installation*, for an installation and configuration checklist.

To configure the Usher components, refer to the *Usher Help*. 
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 chapter describes the following procedures:

• Request an Activation Code, page 121

• Activate your installation, page 123

For answers to commonly asked questions about server activation, see Server Activation FAQ, page 124.

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 UNIX and 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 123.

1 Open MicroStrategy License Manager:

- **Windows**: From the Start menu, point to Programs, then MicroStrategy Tools, and then select License Manager. License Manager opens.

- **UNIX/Linux**: License Manager can be run in GUI mode or command line mode.
  
  — **GUI**: In a UNIX or 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 bin and type ./mstrlicmgr, then press ENTER. The MicroStrategy License Manager opens in GUI mode.

  — **Command line**: In a UNIX or 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 bin and type ./mstrlicmgr -console, then press 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 to continue to the next page.

3 Select the Generate Activation File and Request Activation Code option and click Next to continue to the next page.

4 Enter the characteristics of your server installation and click Next to continue to the next page.

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 to continue to the next page. 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 to continue to the next page. 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.

1. Open MicroStrategy License Manager:
   - **Windows**: From the Start menu, point to Programs, then MicroStrategy Tools, and then select License Manager. License Manager opens.
   - **UNIX/Linux**: License Manager can be run in GUI mode or command line mode.
     - **GUI**: In a UNIX or 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 bin and type ./mstrlicmgr, then press ENTER. The MicroStrategy License Manager opens in GUI mode.
     - **Command line**: In a UNIX or 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 bin and type ./mstrlicmgr -console, then press 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. Refer to the License Manager command line prompts to guide you through the steps to activate your installation. For more information specific to activating 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 to continue to the next page.
   - 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 to activate your software installation.

4. A verification message is displayed, click OK to close it.

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 48* 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](http://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, LPAR ID, number of pooled CPUs, SMT used in AIX

- 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](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
- AIX: Machine ID
- Solaris: Host ID
- RedHat Linux: An identifier generated from a one-way hash of the network interface card MAC address
- HP-UX: Exposes a unique Machine ID that can be obtained from the Operating System command line, using the following command:

  `getconf CS_MACHINE_IDENT`
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.
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 UNIX or 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 UNIX and Linux, see Chapter 11, 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 analysis of your 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 422.*

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 provided 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 134.

• 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 179*).

**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 DataDirect 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, Solaris, AIX, HP-UX, and Redhat or SUSE Linux that are certified for Intelligence Server and different DBMS types, see *Certified ODBC drivers for MicroStrategy Intelligence Server, page 41*.

See the *MicroStrategy Readme* file for details about supported and certified ODBC drivers. To access the *MicroStrategy Readme*:

- On Windows: From the **Start** menu, point to **Programs**, then to **MicroStrategy Documentation**, and then choose **ReadMe**.
- On UNIX/Linux: From the UNIX/Linux File Manager, browse to **INSTALL_PATH**, where **INSTALL_PATH** is the directory that you specified as the install directory during installation. Double-click the **ReadMe.htm** file.

**Note the following:**

- 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 and recommended database connection settings for them. MicroStrategy ODBC drivers only work with MicroStrategy products.

**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 UNIX and Linux

MicroStrategy components require 64-bit drivers to achieve ODBC connectivity.

The ODBC driver manager and support libraries are usually installed in the following default directories:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Default Location for ODBC Driver Manager Libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris</td>
<td>INSTALL_PATH/lib</td>
</tr>
<tr>
<td>AIX</td>
<td>INSTALL_PATH/lib</td>
</tr>
<tr>
<td>HP-UX</td>
<td>INSTALL_PATH/lib</td>
</tr>
<tr>
<td>RedHat and SUSE Linux</td>
<td>INSTALL_PATH/lib</td>
</tr>
</tbody>
</table>

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 387.

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 136).

- 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 137).
Be aware of the following:

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. If you are creating a DSN:
   - On Windows, perform the following steps:
     - Log in to the system as an administrator.
     - From the Start menu, point to Programs, then MicroStrategy Tools, and then select Connectivity Wizard. The Welcome page of the Connectivity Wizard opens.
   - On UNIX or Linux using the Connectivity Wizard interface, perform the following steps:
     - From a UNIX/Linux console window, browse to HOME_PATH, where HOME_PATH is the directory that you specified as the home directory during installation.
     - Browse to the folder bin and type ./mstrconnectwiz, and then press ENTER. The Welcome page of the Connectivity Wizard opens.
   - On UNIX or Linux from the command line, then perform the following steps:
     - From a UNIX/Linux console window, browse to HOME_PATH, where HOME_PATH is the directory that you specified as the home directory during installation.
     - 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 you can skip the rest of this procedure. For detailed steps on how to use the command line version of this tool, see Creating a DSN for a data source, page 354 in Chapter 11, Configuring MicroStrategy Using Command Line Tools.

2 Click Next. A list of database drivers is displayed. The list available for Windows is different than the list available for UNIX/Linux. For a list of the available ODBC drivers for Windows, Solaris, AIX, HP-UX, and Redhat or SUSE Linux that are certified for Intelligence Server and different DBMS types, see Certified ODBC drivers for MicroStrategy Intelligence Server, page 41.

3 Select a database driver with which to create a DSN and click Next. The Driver Details page opens.

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.

4 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 387.

5 Click Test to verify the connection. The Test Connection dialog box opens.

6 Enter the User Name and Password to connect to the database.

7 Click Connect to test and verify the connection. 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.

8 Click Close, and then Finish to create the new DSN.

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.

9 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. A list displays all the existing system data sources and their associated drivers.

   - To view all the installed ODBC drivers, click the Drivers tab.


5. Select the desired driver and click Finish. A driver setup dialog box is displayed.

   - It is 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 387*.

7. Click OK to create a new DSN.

**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.
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, UNIX, and Linux to test and troubleshoot connectivity to databases, create and execute SQL commands through ODBC, and run scripts.

### Prerequisites

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, point to Programs, then MicroStrategy Tools, and then choose DB Query Tool.
   - On Windows from the command line, perform the following steps:
     - From the Windows Start menu, select Run. The Run dialog box opens.
     - In the Open drop-down list, type cmd and click OK. A command prompt opens.
     - Type todbcx.exe and press 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 11, Configuring MicroStrategy Using Command Line Tools.
   - On UNIX or Linux using the DB Query Tool interface, perform the following steps:
— In a UNIX/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 ./mstrdbquerytool, then press ENTER.

• On UNIX or Linux from the command line, perform the following steps:

  — In a UNIX/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 ./mstrtdbcx, then press 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 11, 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 to close the MicroStrategy DB Query Tool.

The DB Query Tool includes many useful features not discussed here. Refer to the DB Query Tool Online Help for details.

**Initial MicroStrategy configuration**

The MicroStrategy Configuration Wizard automates much of the configuration process, prompting you only when critical 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 164.

You can also use the Configuration Wizard to:

- Configure MicroStrategy Health Center, which can help you prevent, diagnose, and fix problems in your MicroStrategy system. Health Center detects known problems and provides an immediate solution to many of them. For steps on how to configure Health Center using the Configuration Wizard, see the System Administration Guide.

- 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 Operations 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 UNIX and 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 131). For a list of certified metadata platforms, see the MicroStrategy Readme.

- In a UNIX or 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 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 6, Deploying MicroStrategy Web, Mobile Server, and Operations Manager.

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**

1. If you are configuring MicroStrategy on:
   - Windows, then perform the following step:
     - From the **Start** menu, point to **Programs**, then **MicroStrategy Tools**, and then choose **Configuration Wizard**. The Configuration Wizard opens. Continue to the steps provided in *To select a configuration task*, page 144.
   - Windows from the command line, then perform the following steps:
     - From the Windows **Start** menu, select **Run**. The Run dialog box opens.
     - In the **Open** drop-down list, type `cmd` and click **OK**. A command prompt opens.
     - Type `macfgwiz` and press **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 164.
   - UNIX or Linux using the Configuration Wizard interface, then perform the following steps:
     - From a UNIX/Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory that you specified as the home directory during installation.
— Browse to the folder bin and type ./mstrcfgwiz, then press ENTER. The Configuration Wizard opens. Continue to the steps provided in To select a configuration task, page 144.

• UNIX or Linux from the command line, then perform the following steps:
  — From a UNIX/Linux console window, browse to HOME_PATH, where HOME_PATH is the directory that you specified as the home directory during installation.
  — Browse to the folder bin and type ./mstrcfgwiz-editor, then press 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 341.

To select a configuration task

1. 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 145.

   • 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 154.

   • 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 Operations Manager Guide.

   • 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 160.

   • Upgrade existing environment to MicroStrategy Analytics 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.

- **Health Center Configuration**: You can use the Configuration Wizard to configure MicroStrategy Health Center, which can help you prevent, diagnose, and fix problems in your MicroStrategy system. Health Center detects known problems and provides an immediate solution to many of them. For steps on how to configure Health Center using the Configuration Wizard, see the System Administration Guide.

The remainder of this chapter 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.

For steps to create metadata, History List, and statistics repositories, see:

- *Creating a metadata repository*, page 146
- *Creating a History List repository*, page 149
- *Creating statistics and Enterprise Manager repositories to maintain and monitor system activity*, page 151

As you complete the configuration process, messages may be displayed. For details on system messages displayed during the configuration process, see Configuration messages, page 154.

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 UNIX and Linux. The steps to perform these two configuration options are provided in the sections listed below:

- *Configuring MicroStrategy with a response file*, page 164
**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 you are upgrading your metadata from a previous version of MicroStrategy rather than creating a brand new metadata, see the [Upgrade Guide](#).
Prerequisites

- 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 146 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 436 to prepare for or avoid specific scenarios that can cause errors.

To create a metadata repository

1 Open the MicroStrategy Configuration Wizard. To do this, see To configure MicroStrategy through the Configuration Wizard, page 143.

2 Select Metadata, History List and Statistics Repository Tables, and click Next. The Repository Configuration: Repository Types page opens.

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 136.

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.
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 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.

To specify a metadata table prefix and complete metadata repository creation

Click Advanced. Options to specify a table prefix and a SQL script to create metadata tables are displayed.

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.

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. For more information on the default SQL scripts, see SQL scripts, page 153.

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 To review and save your metadata configuration, page 148 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 149 and Creating statistics and Enterprise Manager repositories to maintain and monitor system activity, page 151.

- 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.

To review and save your metadata configuration

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. For information on running the Configuration Wizard with a response file, see *Configuring MicroStrategy with a response file, page 164*.

12 Click **Finish** to apply the configuration and create the metadata repository. The summary information is updated as the configurations are completed, providing a way to track the progress of the configurations.

### 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 UNIX or 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 179*.

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.

### Prerequisites

- 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 146* 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 146*.

- 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 436* to prepare for or avoid specific scenarios that may cause errors.
**To create a History List repository**

1. Open the MicroStrategy Configuration Wizard. To do this, see *To configure MicroStrategy through the Configuration Wizard, page 143.*


3. Select the **History List Tables** check box and click Next. The Repository Configuration: History List tables page opens.

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 136.*

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.

**To specify a History List table prefix and complete History List repository creation**

7. Click **Advanced**. Options to specify a table prefix and a SQL script to create History List tables are displayed.

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 179.*

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 153.*

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. You can complete your configuration as described in *To review and save your History List configuration, page 151* below.

- 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 151.*

**To review and save your History List configuration**

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 164.*

12 Click **Finish** to apply the configuration and create the History List repository. 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 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 179.*

**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 *Supplemental Reference for System Administration.*
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 UNIX or 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 179.

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.

Prerequisites

- 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 146 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 146 or Creating a History List repository, page 149, 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 436 to prepare for or avoid specific scenarios that may cause errors.

To create statistics and Enterprise Manager repositories

1. Open the MicroStrategy Configuration Wizard. To do this, see To configure MicroStrategy through the Configuration Wizard, page 143.


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 136*.

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** 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.

**To complete statistics and Enterprise Manager repositories creation**

7 Click **Advanced**. Options to specify a SQL script to create statistics tables are displayed.

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 153*.

9 Click **Next**. The Summary page opens.

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 164*.

11 Click **Finish** to apply the configuration and create the statistics repository. 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 179*.

**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.

For steps to set up Intelligence Server, see *To set up MicroStrategy Intelligence Server, page 155* below.

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 UNIX and Linux. The steps to perform these two configuration options are provided in the sections listed below:

- *Configuring MicroStrategy with a response file, page 164*

**Prerequisites**

- 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. To do this, see *To configure MicroStrategy through the Configuration Wizard, page 143*.

2. Select *Configure Intelligence Server*, and click *Next*. The Server Configuration: Metadata Connection page opens.

3. From the DSN drop-down list, select a DSN for the data source that stores the metadata and specify a *User Name* and *Password*. If the password to the metadata has changed in the database, ensure that this is reflected in the password that you provide.

   You can also use the options listed below:
   
   - *New* to create a new DSN (see *Creating a DSN for a data source, page 136*)
   - *Advanced* if you want to specify a metadata table prefix, which is an identifier stored in the project metadata associated with a table or tables and is used to generate SQL

   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. In the Server Configuration: 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. Refer to the System Administration Guide for details about passwords and other user management information.

6 Click **Next**. The Server Configuration: Server Definitions page opens.

**To create, use, or delete a server definition**

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 on the Intelligence Server Configuration Editor, refer to the System Administration Guide.
  
a  In the **Name** field, type a name to identify the server definition.

  b Select the **Use as the active server definition** check box to define Intelligence Server to use the new server definition when Intelligence Server starts.

  c Click **Next**. The Server Configuration: Settings page opens, described in *To define the Intelligence Server port number and other settings, page 157* 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.
  
a  From the Existing Server Definitions pane, select a server definition to use.

  b Click **Next**. The Server Configuration: Settings page opens, described in *To define the Intelligence Server port number and other settings, page 157* 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.
  
a  From the Existing Server Definitions pane, select a server definition to delete.

  b Click **Next**. The Summary page opens, described in *To complete Intelligence Server configurations, page 159* below.
To define the Intelligence Server port number and other settings

8 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 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 435.

- **Register Intelligence Server as a Service**: This option is only available if you are configuring Intelligence Server on a UNIX or Linux machine, and you have root access and permissions to the UNIX or Linux machine that Intelligence Server is installed on. Select this check box to register Intelligence Server as a service.

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 UNIX or Linux, you must stop the service, and then use the *mstrctl unregister* command line tool. The syntax is *mstrctl -s IntelligenceServer us*, where *IntelligenceServer* is the name of a server definition. For information on 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.

9 Click **Next**. The SSL Configuration Page opens.
To encrypt the communications between Developer and Intelligence Server

10 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 use the SSL protocol. Clear this check box to disable the use of the SSL protocol for Intelligence Server and Developer communications.

Select this check box to enable the use of the SSL protocol for Intelligence Server and Developer communications. Be aware that you must ensure the following prerequisites are met to enable the SSL protocol:

- You must have access to the SSL certificate you created for Intelligence Server.
- You must have the private key file that you created while requesting a certificate for Intelligence Server. For information on creating a private key and obtaining an SSL certificate, refer to the [System Administration Guide](https://example.com).

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](https://example.com).

11 Click Next. The Statistics Configuration page opens.

To specify the default statistics repository

12 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. You can also enable basic statistics logging for projects:

- **Make this my default Statistics Database Instance for the local Intelligence Server metadata**: Select this check box to define which statistics repository to use for recording statistics. If you clear this check box, a default statistics database instance is not defined for your Intelligence Server.
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.

- **User Name**: Type the database user name for the user that can connect to the statistics data source.

- **Password**: Type the password for the user that can connect to the statistics data source.

- **Enable Basic Statistics (For newly created projects and existing projects not logging statistics)**: Select this check box to start logging basic statistics for new projects and any projects that are not currently logging statistics. You must use the Project Configuration Editor available in MicroStrategy Developer to:
  - Enable additional statistics for a project.
  - Enable basic and additional statistics for a project if you cleared this check box.

**13** Click **Next**. The Summary page opens.

**To complete Intelligence Server configurations**

**14** Review the summary information.

You can click **Save** to save the configuration as a response (.ini) file to configure Intelligence Server 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 164.*

**15** Click **Finish** to apply the Intelligence Server configuration. The summary information is updated as the configurations are completed, providing a way to track the progress of the configurations.

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, refer to 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 160: Direct project sources that connect directly to the metadata through ODBC. You cannot create a direct project source on UNIX or Linux.
- Creating a server (three-tier) project source, page 161: 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 UNIX and Linux. The steps to perform these two configuration options are provided in the sections listed below:

- Configuring MicroStrategy with a response file, page 164

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 UNIX or Linux.

Prerequisites

- 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. To do this, see To configure MicroStrategy through the Configuration Wizard, page 143.
2 Select **Project Sources**, and click **Next**. The Project Source Creation: Name page opens.

3 In the **Project Source Name** field, type a name for the project source.

4 Under **Connection Type**, select **Direct (2-tier)**, and click **Next**. The Project Source Creation: Metadata Location page opens.

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**.

   You can also click **New** to create a new DSN (see *Creating a DSN for a data source, page 136*) and click **Advanced** to specify a metadata table prefix if necessary.

6 Click **Next**. The Project Source Creation: Authentication page opens.

7 Select the authentication mode for the project source. For information on the available authentication modes, see the **Authentication modes, page 163**.

8 Click **Next**. The Summary page opens.

9 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 164*.

10 Click **Finish** to create the project source. The summary information is updated as the configurations are completed, providing a way to track the progress of the configurations.

### 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.

### Prerequisites

- 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. To do this, see To configure MicroStrategy through the Configuration Wizard, page 143.

2. Select Project Sources and click Next. The Project Source Creation: Name page opens.

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. The Project Source Creation: Metadata Location page opens.

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.


9. Select the authentication mode for the project source. For information on the available authentication modes, see the Authentication modes, page 163.

10. Click Next. The Summary page opens.

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 164.

12. Click Finish to create the project source. The summary information is updated as the configurations are completed, providing a way to track the progress of the configurations.
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:

- **Network login ID: Windows authentication, page 163**
- **Login ID and password entered by the user: Standard authentication, page 163**
- **Guest account: Anonymous authentication, page 163**
- **LDAP authentication, page 164**
- **Login ID and password entered by the user for the warehouse: Database authentication, page 164**
- **Integrated authentication, page 164**

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, UNIX, and Linux.

The Configuration Wizard can also be used to perform MicroStrategy upgrades, configure Health Center, 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 configure Health Center with a response file, see the System Administration Guide.
• For steps to create an Enterprise Manager project with a response file, see the Operations 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 165 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 UNIX and Linux machines. For steps to create and use a response file as well as perform other configurations using command line tools in UNIX and Linux, see the Configuring MicroStrategy with a response.ini file, page 357 section in Chapter 11, 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 143.

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:
   • Creating metadata, History List, and statistics repositories, page 145
   • Setting up MicroStrategy Intelligence Server, page 154
   • Creating project sources, page 160

3 Once you reach the Summary page for a configuration, click Save. The Save dialog box opens.

4 Specify a name and location to save the response file, and click Save. You are returned to the Summary page.

5 To also perform the configuration task, click Finish. The summary information is updated as the configurations are completed, providing a way to track the progress of the configurations.

Steps to use a response file to configure MicroStrategy are covered in Using a response file to configure MicroStrategy installations, page 166 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 167.
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 166:* 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 166:* 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 UNIX or 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 UNIX and Linux, see the Configuring MicroStrategy with a response.ini file, page 357 section in Chapter 11, 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 143.
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 UNIX and Linux, see Chapter 11, 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, page 167*
- *Setting up MicroStrategy Intelligence Server, page 169*
- *Creating and configuring project sources, page 173*

**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>CreateMDFiles=</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>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>• UNIX/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>• UNIX/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>• UNIX/Linux:/INTELLIGENCE_SERVER_INSTALL_PATH/statistics_DB2.sql.</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>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
<tr>
<td>TBName=</td>
<td>Defines the tablespace name to be created in the database. 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>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</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 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>DSNName=</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>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</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>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.</td>
</tr>
<tr>
<td>RegisterAsService=</td>
<td>This option is only available on Intelligence Servers running on UNIX or 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>
<tr>
<td></td>
<td>• 1: Registers Intelligence Server as a service. Performing this task requires a UNIX or Linux login with root level access and privileges.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not register Intelligence Server as a service.</td>
</tr>
<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>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</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 private key for the SSL certificate.</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>
</tbody>
</table>
|                       | • 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,
### Options and Description

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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>DefaultStatisticsPrefix</td>
<td>Defines a prefix for statistics repository tables used by the server definition.</td>
</tr>
<tr>
<td>DefaultStatisticsBasicStats</td>
<td>Defines whether to enable basic statistics, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Enables the logging of basic statistics for new projects and any projects that are not now logging statistics.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not enable the logging of basic statistics for new projects and any projects that are not now logging statistics.</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>[Client]</td>
<td>In this section you can configure the project source name. You can have more than one [Client] section. Additional client sections can be included as [Client1],[Client2], 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 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>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DataSource=</td>
<td>Defines the name of the new project source to create.</td>
</tr>
</tbody>
</table>
| ConnType=   | Defines the database connection type for a project source. The following connection types are supported:  

- 2: Connects a project source to the metadata using an ODBC DSN (Windows only).  
- 3: Connects a project source to the metadata through a MicroStrategy Intelligence Server (three-tier). |
| DSN=       | If using connection type 2 (ConnType=2), defines the name of the ODBC database. |
| UserName=  | If using connection type 2 (ConnType=2), defines the user name to connect to the ODBC database. |
| UserPwd=   | If using connection type 2 (ConnType=2), defines the password to log in to the database. |
| ServerName=| If using connection type 3 (ConnType=3), defines the name of the MicroStrategy Intelligence Server to connect to. |
| Port=      | 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. |
| Authentication= | The following authentication modes are supported:  

- 1: Standard or login ID and password entered by the user  
- 2: Network login ID (Windows authentication)  
- 8: Guest account (Anonymous authentication)  
- 16: LDAP authentication  
- 32: Database login ID and password (database authentication)  
- 128: Integrated authentication  

For information on the available authentication modes, see the Authentication modes, page 163. |
| 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. |
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.

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 179, Creating a database connection, page 181, and Creating a database login, page 186.

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, Hyperion 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.

The database instances that you create are separated into two categories:
• *SQL data warehouses database instances, page 176*
• *MDX cube database instances, page 177*

**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, Hyperion 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.

![Database Instances Editor](image)

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. Expand Administration, then expand Configuration Managers, and then select Database Instances.

3. From the File menu, point to New, and then select 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 *To create a database connection, page 183*.
  
  — **Use 3.x ODBC Calls**: The support of ODBC 3.x is introduced in MicroStrategy 9.0. You should use this option if your database management system supports ODBC 3.x.

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 181* 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 187*.

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.
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.

- **HiveThrift Connector database instances**
  - General: Specifies the database connection name, the default database login, and additional connection information that you must provide. For more information on defining connection information to Hadoop Hive distributions by using a database connection, see ODBC Driver for Red Brick for UNIX/Linux, page 402.
Installation and Configuration Guide

Advanced: Specifies the database driver mode, driver execution mode, and other miscellaneous warehouse connection settings.

The steps below show you how to create a database connection for a relational database instance. For information on creating a database connection for MDX cube sources, refer to the MDX Cube Reporting Guide.

Prerequisites

- A database instance has been created, as described in Creating a database instance, page 179.

To create a database connection

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 functionality is only applied if the database instance is defined to use 2.0 ODBC calls, which is included in the steps To create a database instance, page 180.

  The MicroStrategy Readme recommends settings for ODBC drivers and whether to use the extended fetch feature.

- **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 lifetime** *(sec)*: Defines the amount of time an active database connection can remain open and cached on Intelligence Server to be re-used for additional jobs. You must also set the Connection idle timeout, described below, to a value greater than zero for database connections to be used by more than a single job. If a job requires a database connection to be open past its connection lifetime, the job is first allowed to complete, and then the database connection is dropped upon job completion.

    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, the lifetime of a database connection is unlimited, which means it remains on Intelligence Server memory until the database connection is manually deleted or Intelligence Server is restarted. This can cause the memory resources to be reserved by the database connection for a potentially long time, and therefore it is recommended that you set finite limits on the connection lifetime.

    When defining the connection lifetime, you should determine whether the data source for the database connection also enforces connection lifetimes. Most databases enforce some type of limit on a connection lifetime. You should define the connection lifetime in MicroStrategy to be less than any limits on connection lifetimes for the data source. This is to avoid the scenario that the data source ends the database connection before MicroStrategy can complete the processing being done for that same database connection.

  — **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.

    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 186* 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 187.

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.

Prerequisites

• A database instance has been created, as described in Creating a database instance, page 179.

• A database connection has been created, as described in Creating a database connection, page 181.

To create a database login

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.

Prerequisites

   • A database instance has been created, as described in Creating a database instance, page 179.
• A database connection has been created, as described in Creating a database connection, page 181.

• A database login has been created, as described in Creating a database login, page 186.

To create a connection map

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 49* in *Chapter 1, Planning Your Installation*, for an installation and configuration checklist.
DEPLOYING MICROSTRATEGY WEB, MOBILE SERVER, AND OPERATIONS MANAGER

This chapter describes the procedure to deploy a project to your user community using MicroStrategy Web, Mobile Server, and Operations Manager. The process of deploying the ASP.NET version of MicroStrategy Web, MicroStrategy Mobile Server, or Operations Manager on Windows with Microsoft Internet Information Services (IIS) is explained in detail.

Steps to deploy MicroStrategy Web (JSP), MicroStrategy Mobile Server (JSP), or MicroStrategy Operations Manager (JSP) in a UNIX/Linux and Windows environment with various Web and application servers are also explained in detail. MicroStrategy Web (JSP), Mobile Server (JSP), and Operations Manager (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.

This chapter has the following sections:

- Deploying with IIS (Windows), page 191

Deploying with IIS is the only setup given for the ASP.NET versions of MicroStrategy Web, Mobile Server, and Operations Manager. The other deployment procedures use the JSP, platform-independent versions, which can be deployed with different Web and application servers.
• General steps to deploy MicroStrategy JSP applications, page 194
• Deploying with WebLogic and Apache (Solaris), page 198
• Deploying with WebSphere and IBM HTTP Server (AIX), page 212
• Deploying with Oracle Glassfish Server (Solaris), page 222
• Deploying with Tomcat (Windows), page 230
• Deploying with Tomcat (Linux), page 236
• Deploying with SAP NetWeaver (Windows), page 241
• Deploying with Oracle 10g (Windows), page 245
• Deploying with JBoss (Windows), page 248
• Administering your MicroStrategy Web deployment, page 254
• Using absolute paths to share configuration files
• Configuring third-party data sources for importing data
• Configuring your MicroStrategy installation, page 257

Deploying with IIS (Windows)

Microsoft IIS can be used to deploy MicroStrategy Web, MicroStrategy Mobile Server, or MicroStrategy Operations Manager:

Deploying MicroStrategy Web

The ASP.NET version of MicroStrategy Web can be deployed with IIS only on Windows.

Prerequisites

• 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 34.

• 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:

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

Prerequisites

- 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*, page 34.
• 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:
  - 32-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. On the Windows **Start** menu, point to **Programs**, then to **MicroStrategy Tools**, and then select **Mobile Administrator**. The MicroStrategy Mobile Server Administrator page opens. This is the page where you connect MicroStrategy Mobile Server to the Intelligence Server.

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 configuration, see the [MicroStrategy Mobile Design and Administration Guide](https://www.microstrategy.com/docs/microstrategy-mobile-design-and-administration-guide).

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**.
Deploying Operations Manager

The ASP.NET version of MicroStrategy Operations Manager can only be deployed with IIS only on Windows.

Prerequisites

- For information on supporting and IIS 7.x, see Supporting IIS 7.0.x or IIS 7.5.x as a web server for MicroStrategy Web or Mobile Server, page 34.

- You must have administrative privileges to deploy MicroStrategy Operations Manager. 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 Operations Manager. By default, this folder is stored in the following directory location:

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

To connect MicroStrategy Operations Manager to your Intelligence Server

1. On the Windows Start menu, point to Programs, then to MicroStrategy Tools, and then select Operations Manager. Operations Manager opens.

2. Click Configuration ( ) and then click Create Environment to connect to the MicroStrategy environment that is to be monitored and managed using Operations Manager. This includes connecting to an Intelligence Server, along with any MicroStrategy Web Servers or Mobile Servers that are associated with the Intelligence Server. For steps to create an environment, see the Operations Manager Guide.

General steps to deploy MicroStrategy JSP applications

After you have installed MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (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.
<table>
<thead>
<tr>
<th>Application</th>
<th>High-Level Deployment Steps</th>
</tr>
</thead>
</table>
| 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.  
   ![Note](image) You must perform extra configuration steps to allow graphs to support non-Western European fonts on MicroStrategy Web (JSP) for a UNIX system. For more information, see Graph and document support of non-Western European fonts, page 434 of Appendix C, 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 Design and Administration Guide.  
7. Start working with the application. |
| MicroStrategy Operations Manager (JSP) | 1. Log on to the application server by using the proper user name and password.  
3. Choose the desired deployment method and follow the deployment procedure.  
4. Log on to the MicroStrategy Operations Manager console to connect to the MicroStrategy environment that is to be monitored and managed using Operations Manager. For steps to create an environment, see the Operations Manager Guide. |

**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>
</table>
| MicroStrategy Web (JSP)     | • 32-bit Windows environments: C:\Program Files\MicroStrategy\WebJSP  
| MicroStrategy.war           | • 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\WebJSP  
|                             | • UNIX and Linux environments: INSTALL\_PATH/WebUniversal  
|                             | Where INSTALL\_PATH is the directory you specified as the MicroStrategy install directory during installation. |
| MicroStrategy Mobile Server (JSP) | • 32-bit Windows environments: C:\Program Files\MicroStrategy\Mobile Server JSP  
| MicroStrategyMobile. war; | • 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Mobile Server JSP  
|                             | • UNIX and Linux environments: INSTALL\_PATH/Mobile  
|                             | Where INSTALL\_PATH is the directory you specified as the MicroStrategy install directory during installation. |
| MicroStrategy Operations Manager (JSP) | • 32-bit Windows environments: C:\Program Files\MicroStrategy\Operations Manager_JSP  
| MicroStrategyOM.war         | • 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Operations Manager JSP  
|                             | • UNIX and Linux environments: INSTALL\_PATH/  
|                             | Where INSTALL\_PATH is the directory you specified as the MicroStrategy install directory during installation. |

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 196.

**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>
<tr>
<td>\html</td>
<td>Supporting files</td>
</tr>
<tr>
<td>\images</td>
<td>All image files</td>
</tr>
<tr>
<td>\import</td>
<td>Sample files for Data Import analysis</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>\css</td>
<td>Supporting css files</td>
</tr>
<tr>
<td>\html</td>
<td>Supporting files</td>
</tr>
<tr>
<td>\images</td>
<td>All image files</td>
</tr>
<tr>
<td>\import</td>
<td>Sample files for Data Import analysis</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
MicroStrategy Mobile Server (JSP), and MicroStrategy Operations Manager (JSP).

This section includes the following information:

- **WebLogic paths and folder locations, page 199**: Default folder structure for each version of WebLogic.
- **Preconfiguration information, page 199**: Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP).
- **Deploying MicroStrategy Web, Mobile Server, or Operations Manager, page 201**: Instructions for deploying the application.
- **Re-deploy the application, page 208**: Instructions for re-deploying the application.
- **Performance-based setup information, page 208**: 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 10.3: WEBLOGIC_HOME/user_projects/domains/mydomain/

**Note the following:**

- **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 includes the following sections:
•  *Locating the WAR file, page 195*

•  *Setting up Apache Web server to proxy requests to WebLogic, page 200*

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>
<tr>
<td>JDK</td>
<td>Oracle JDK 1.6.0 or 1.7.0</td>
</tr>
</tbody>
</table>

Note the following:

- 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 208.*

**Deploying MicroStrategy Web, Mobile Server, or Operations Manager**

When your machine has been configured with the necessary settings, you can deploy the JSP version of MicroStrategy Web, Mobile Server, or Operations Manager with Apache and WebLogic. This involves the following steps:

1. **Deploying automatically (development mode), page 201.**
   - or -
   **Deploying manually (production mode), page 204.**

2. **Configuring administrative access to MicroStrategy JSP applications, page 218.**

3. **Launching the project, page 208.**

   The *Performance-based setup information, page 208* 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, Mobile Server, or Operations Manager using one of the following deployment methods:

- The automatic deployment feature is the easiest and fastest way. See *Deploying automatically (development mode), page 201.* 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 204.*

**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 195.

2 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. This step is optional.

   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

3 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:
   a 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.

   b Open the web.xml file located in the /WEB-INF directory.

   c 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.

   d Save the web.xml file.

   e 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

4 Transfer the WAR file to the following directory:

```
/WEBLOGIC_MYDOMAIN_FOLDER/autodeploy
```
The application is automatically deployed. To add and connect to an Intelligence Server, see Configuring administrative access to MicroStrategy JSP applications, page 206.

To increase the performance of MicroStrategy Web JSP, you can configure additional settings after deployment. For more information, see Performance-based setup information, page 208.

To automatically deploy MicroStrategy JSP applications from an exploded directory

The WAR file must be uncompressed by the same Solaris user who started the application.

1. Locate the WAR file for your MicroStrategy JSP application, as described in Locating the WAR file, page 195.

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:

   # rm FileName.war

   Where FileName is the name of the WAR file for your MicroStrategy JSP application.

6. Move the folder to the autodeploy folder with the following commands:

   # cd..

   # mv context_folder /WEBLOGIC_MYDOMAIN_FOLDER/autodeploy

The application is automatically deployed. To add and connect to an Intelligence Server, see Configuring administrative access to MicroStrategy JSP applications, page 206.
To increase the performance of MicroStrategy Web (JSP), you can configure additional settings after deployment. For more information, see Performance-based setup information, page 208.

Deploying manually (production mode)

With manual deployment you can deploy MicroStrategy JSP applications 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 to access the required folders to perform any necessary configurations to files in the exploded directory.

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 195.

To modify the web.xml file for multiple MicroStrategy deployments

2 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 as described below:

   a Unzip the WAR file by using the following command:

   ```sh
   #jar -xvf FileName.war
   ```

   Where FileName is the name of the WAR file for your MicroStrategy JSP application.

   b Open the web.xml file located in the /WEB-INF directory.

   c 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.

   d Save the web.xml file.

   e Zip the WAR file by using the following command:

   ```sh
   #jar -cvf FileName.war *
   ```
Where FileName is the name of the WAR file for your MicroStrategy JSP application.

**To deploy the WAR file**

3 Transfer the WAR file to the /WEBLOGIC_MYDOMAIN_FOLDER/autodeploy directory.

4 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.

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.

6 To complete this operation, see *Configuring from the WebLogic Server Administration Console*, page 206.

---

**To manually deploy MicroStrategy JSP applications from the exploded directory**

1 Locate the WAR file for your MicroStrategy JSP application, as described in *Locating the WAR file*, page 195.

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:

   #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.

6 To complete this operation, see *Configuring from the WebLogic Server Administration Console*, page 206 below.
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, Mobile Server, or Operations Manager 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), or Operations Manager (JSP), see Configuring administrative access to MicroStrategy JSP applications, page 206.

To increase the performance of MicroStrategy Web (JSP), you can configure additional settings after deployment. For more information, see Performance-based setup information, page 208.

Configuring administrative access to MicroStrategy JSP applications

Before you start MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (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, MicroStrategy Mobile Server Administrator, and Operations Manager for each deployment method.

   The servlet names are case-sensitive. Make sure to 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.

<table>
<thead>
<tr>
<th>Deployment Method</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic deployment</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, context_folder_Web is the name of the folder where the Web (JSP) application was exploded and IPaddress 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, context_folder_Mobile is the name of the folder where the Mobile Server application was exploded and IPaddress is the IP address of your machine.</td>
</tr>
<tr>
<td></td>
<td>• For Operations Manager (JSP): <a href="http://IPaddress:7001/context">http://IPaddress:7001/context</a>_</td>
</tr>
<tr>
<td>Deployment Method</td>
<td>Address</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>folder_OM/servlet/mstrOM</td>
</tr>
</tbody>
</table>

In the URL listed above, context_folder_OM is the name of the folder where the Operations Manager application was exploded and IPAddress is the IP address of your machine.

<table>
<thead>
<tr>
<th>Manual deployment</th>
<th>Access the Administrator page from a browser using this address:</th>
</tr>
</thead>
<tbody>
<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, IPAddress is the IP address of your machine. Replace the Web_name variable with the name you specified in the deployed name field when configuring Web (JSP) from WebLogic Server Administration Console, for example, MyWebApp.</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, IPAddress is the IP address of your machine. Replace the Mobile_name variable with the name you specified in the deployed name field when configuring Mobile Server from WebLogic Server Administration Console, for example, MyMobileApp.</td>
</tr>
<tr>
<td></td>
<td>• For Operations Manager (JSP): <a href="http://IPAddress:7001/OM_name/servlet/mstrOM">http://IPAddress:7001/OM_name/servlet/mstrOM</a></td>
</tr>
<tr>
<td></td>
<td>In the URL listed above, IPAddress is the IP address of your machine. Replace the OM_name variable with the name you specified in the deployed name field when configuring Operations Manager from WebLogic Server Administration Console, for example, OperationsManagerApp.</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 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 Design and Administration Guide**. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

- If you are deploying MicroStrategy Operations Manager (JSP), you can begin to monitor and manage your MicroStrategy server systems. For steps to create an environment, see the **Operations Manager Guide**.
4 For MicroStrategy Web (JSP) deployments, launch the MicroStrategy Web (JSP) project, as described in *Launching the project, page 208*. 

**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: &lt;br&gt;<a href="http://IPaddress:7001/context_folder/servlet/mstrWeb">http://IPaddress:7001/context_folder/servlet/mstrWeb</a> &lt;br&gt;where <em>context_folder</em> is the name of the folder where the application was exploded and <em>IPaddress</em> is the IP address of your machine.</td>
</tr>
<tr>
<td>Manual deployment</td>
<td>Access MicroStrategy Web (JSP) from a browser using the address: &lt;br&gt;<a href="http://IPaddress/name/servlet/mstrWeb">http://IPaddress/name/servlet/mstrWeb</a> &lt;br&gt;where <em>IPaddress</em> is the IP address of your machine. Replace the <em>name</em> variable with the name you specified in the deployed <em>name</em> field when configuring from WebLogic Server Administration Console, for example, <em>MyWebApp</em>.</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, page 209*
- *Precompiling JSP files, page 209*
- *Disable/relax auto-reload parameters, page 210*
- *Configuring Apache Web server to serve static files, page 211*
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.

To increase the Java heap size

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.

To precompile the JSP files

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:

   ```
   <jsp-descriptor>  
     :  
     :  
     <jsp-param>  
       <param-name>keepgenerated</param-name>  
       <param-value>TRUE</param-value>  
     </jsp-param>  
     <jsp-param>  
       <param-name>precompile</param-name>  
       <param-value>TRUE</param-value>  
     </jsp-param>  
   </jsp-descriptor>
   ```
Enable/relax auto-reload parameters

To enable/relax auto-reload parameters, complete the following:

• *Set the pageCheckSeconds parameter, page 210*
• *Set the WebLogic Reload Period parameter, page 211*

Each parameter is explained below.

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 198 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 provided below to deploy MicroStrategy Web (JSP), MicroStrategy Mobile Server (JSP), and MicroStrategy Operations Manager (JSP).

This section includes the following information:

- *Preconfiguration information, page 213:* Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP).

- *Deploying MicroStrategy Web, Mobile Server, or Operations Manager, page 213:* Instructions for deploying the application.
• *Performance-based setup information, page 220:* Optional settings to increase the application’s 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 any of these options are of interest to you.

## Preconfiguration information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on your machine. Preconfiguration includes the following step:

• *Locating the WAR file, page 195:* This file is used as part of the steps for *Preparing for the application installation, page 215.*

This section also provides additional setup information for the machine where the application server and Web server are already installed.

This section supports the following configuration. While your configuration 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>AIX 6.1.x or 7.1.x (on IBM POWER Architecture)</td>
</tr>
<tr>
<td>Web server</td>
<td>IBM HTTP Server 7.x or 8.x</td>
</tr>
<tr>
<td>Application server</td>
<td>IBM WebSphere application server 7.x, 8.0.x, or 8.5.x</td>
</tr>
<tr>
<td>JDK</td>
<td>IBM JDK 1.6.0</td>
</tr>
<tr>
<td></td>
<td>You can download these JDKs from the following location:</td>
</tr>
</tbody>
</table>

For information on the version numbers certified and supported by MicroStrategy, see the *MicroStrategy Readme.*

## Deploying MicroStrategy Web, Mobile Server, or Operations Manager

Once your machine has the necessary settings configured, you can deploy MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP) on the WebSphere machine. Deployment involves the following steps:

1. *Launching the WebSphere Administrative Console, page 214*
2. *Starting the WebSphere default application server, page 214*
3. *Installing the Enterprise Application, page 215*
4. *Regenerating plugin-cfg.xml, page 217*
5  **Restarting the application server, page 217**

6  **Configuring administrative access to MicroStrategy JSP applications, page 218**

7  **Launching the project, page 219**

The **Performance-based setup information, page 220** 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 any of these options are of interest to you.

---

**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**

1. 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. A table listing the application servers displays to the right of the navigation tree. This area is the Workspace.
3. Select the box next to the application server to start.
4. Click *Start* above the table.

**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.

**Prerequisites**

- Copy the WAR file (see *Locating the WAR file, page 195*) for the MicroStrategy JSP placation 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, expand *Applications*, then expand *Application Types*, and then select *WebSphere enterprise applications*. A list of enterprise applications is displayed.
2. Click *Install*. Options to specify the path to the new application are displayed.
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** to continue the installation.

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**. The Select installation options page opens. In the screens that follow, you are selecting settings that are used during the installation.

8 Perform the following configuration steps:
   - Select the **Precompile Java Server Pages files** check box.
   - Specify the value for the **Directory to Install Application** as `${APP_INSTALL_ROOT}/DefaultNode`.
   - Specify an **Application Name** of your choice.
   - Ensure that the **Override class reloading settings for Web and EJB modules** check box is cleared.

9 Click **Next**. The Map modules to application servers page opens.

10 Select the **Web Tier** check box and click **Next**. The Map context roots for web modules page opens.

11 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`.

12 Click **Next**. The Summary Page opens.

13 Review the summary and click **Finish**. A message appears stating that the installation and precompilation of JSPs was successful. Save the changes to the master repository.

14 As part of deploying MicroStrategy Web (JSP) or MicroStrategy Mobile (JSP), you can control access to the MicroStrategy Web Administrator page, MicroStrategy Mobile Server Administrator page, and Operations Manager respectively.

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

To regenerate 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 218.

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. A table listing the application servers and an icon indicating their status appears:
   • red: stopped
   • green: started

2. Select the box next to the application server you want to stop, and click Stop. The status icon changes from green to red.

3. Select the application server you want to start and click Start. The application server starts and the status icon changes to green.

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. A list of enterprise applications is displayed, along with icons indicating their status:
   - red: stopped
   - green: started

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 215. For more information, you can 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 215. 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 section Preparing for the application installation, page 215. For example, the default name of the WAR file, which is MicroStrategyMobile.
   - For Operations Manager (JSP):
     http://IPAddress/ContextRootOM/servlet/mstrOM
     In the URL listed above, ContextRootOM is the name you provided for the ContextRoot for Web Module box in the section Preparing for the application installation, page 215. For example, the default name of the WAR file, which is MicroStrategyOM.
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:
   - 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 Design and Administration Guide](https://www.mstrategy.com/support/mobile-design-administration-guide). Creating a configuration completes the steps required to deploy Mobile Server (JSP).
   - If you are deploying MicroStrategy Operations Manager (JSP), you can begin to monitor and manage your MicroStrategy server systems. For steps to create an environment, see the [Operations Manager Guide](https://www.mstrategy.com/support/operations-manager-guide).

4 If you are deploying MicroStrategy Web (JSP), proceed to launch the project. For more information, see *Launching the project, page 219*.

### 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:

   ```bash
   # /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:

   ```bash
   /usr/IBMIHS/bin/apachectl
   ```

2 In a Web browser, specify the following URL:

   ```text
   http://MachineName/ContextRoot/servlet/mstrWeb
   ```

   Alternatively, you can use the IP address of your machine for remote access, as shown below:

   ```text
   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 215.

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.

To uninstall MicroStrategy Web application

1. Expand Applications, then expand Application Types, and then select WebSphere enterprise applications. A list of enterprise applications is displayed.

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). This section explains the following changes:

- Setting the Java heap size, page 220
- Precompiling JSP files, page 221
- Configuring the IBM HTTP Server to serve static files, page 222

**Setting the Java heap size**

You can increase the Java heap size for a given application server by configuring the WebSphere Administrative Console.

To increase the Java heap size

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 500MB, 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** and save your changes.

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.

     ```bash
     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 UNIX/Linux environment. You can use the steps provided below to deploy MicroStrategy Web (JSP), MicroStrategy Mobile Server (JSP), and MicroStrategy Operations Manager (JSP).

- *Preconfiguration information, page 222*: Configuration that must occur before you begin deploying MicroStrategy JSP application.
- *Deploying MicroStrategy Web, Mobile Server, or Operations Manager, page 223*: Instructions for deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP).

**Preconfiguration information**

This section provides the preconfiguration information necessary for deploying MicroStrategy JSP applications on your machine, which includes *Locating the WAR file, page 195*. It also provides additional setup information that needs to be done on the machine on which the application server and Web server are already installed.

While your configuration may vary slightly, for example you may have different versions of these applications, the overall process remains the same:
For information on the version numbers certified and supported by MicroStrategy, see the *MicroStrategy Readme*.

**Deploying MicroStrategy Web, Mobile Server, or Operations Manager**

Once your machine has been configured, you can deploy MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP) with Oracle Glassfish Server 3.1.x.

The deployment involves the steps below, which are explained in detail in the following sections:

1. *Launching the Oracle Glassfish Server Administration Console, page 223*
2. *Deploying your MicroStrategy JSP application, page 224*
3. *Configuring administrative access to MicroStrategy JSP applications, page 226*
4. *Accessing the MicroStrategy JSP application administrative page, page 228*
5. *Connecting to the Web (JSP) project page, page 229*

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.

**Prerequisites**

- 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 195*) 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:

   \( \text{InstallDir/bin} \)

   where \( \text{InstallDir} \) is the directory where you installed Oracle Glassfish Server.

2. Type the following command to start the domain:

   \( \text{asadmin start-domain --domaindir DomainDirectory DomainName} \)

   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:

   \( \text{asadmin start-domain domain1} \)

3. Access the Oracle Glassfish Server Administration Console by typing the following URL:

   \( \text{http://MachineName:PortNumber} \)

   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.

#### Prerequisites

• Save the WAR file (see *Locating the WAR file, page 195*) to the same machine as the Oracle Glassfish Server, or to a location that is accessible to the Oracle Glassfish Server.
Access to the administrative pages for MicroStrategy Web (JSP), Mobile Server (JSP), and Operations Manager (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, Mobile Server, or Operations Manager, page 223.*

**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**. The Applications page is displayed.

5. Click **Deploy**. The Deploy Applications or Modules page is displayed.

6. Select **Local Packaged File or Directory That Is Accessible from GlassFish Server**, and then click **Browse Files**. The Browse Server dialog box opens.

   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.

The URL to access the MicroStrategy Operations Manager (http://IPAddress:PortNumber/ContextRoot/servlet/mstrOM) 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 MicroStrategyOM.

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 to deploy the application.

**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, or Operations Manager. 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:


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, expand Configuration, then server-config, then Security, then Realms, and then select file. The Edit Realm page is displayed.

3. Click Manage Users. The File Users page is displayed.

4. Click New to create a new user. The New File Realm User page is displayed.

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, MicroStrategy Mobile Server Administrator, and Operations Manager.
   - **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 to create the user. You are returned to the File Users page, where you can continue to create and manage users for the administration realm.

7. To apply all of these changes, stop and restart the application server:
   a. In the Administration Console, from the Tree pane on the left, select server (Admin Server). The General Information page is displayed.
   b. 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, Mobile Server Administrator, and Operations Manager. 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 224 and Configuring administrative access to MicroStrategy JSP applications, page 226 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 224*) and assign the security role to the necessary user accounts (see *Configuring administrative access to MicroStrategy JSP applications, page 226*).

### 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 224*. 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://
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 224. For example, you can use the default name of the WAR file, which is *MicroStrategyMobile*. The default port number is 8080.

- For Operations Manager (JSP):

  http://IPAddress:PortNumber/ContextRootMobile/servlet/mstrOM

  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 224. For example, you can use the default name of the WAR file, which is *MicroStrategyOM*. The default port number is 8080.

  The servlet names are case-sensitive. Use the correct case when typing the 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 226.

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 Design and Administration Guide*. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

   - If you are deploying MicroStrategy Operations Manager (JSP), you can begin to monitor and manage your MicroStrategy server systems. For steps to create an environment, see the *Operations Manager Guide*.

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 229 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

1. 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 provided below to deploy MicroStrategy Web (JSP), MicroStrategy Mobile Server (JSP), and MicroStrategy Operations Manager (JSP):

- **Preconfiguration information, page 230**: Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP).
- **Deploying MicroStrategy Web, Mobile Server, or Operations Manager, page 233**: Instructions for deploying, including step-by-step procedures.

Preconfiguration information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on Tomcat on your machine.
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>Windows 2008 Enterprise Edition SP2 (on x64)</td>
</tr>
<tr>
<td>Application server</td>
<td>Tomcat 7.0.x or 8.0.x</td>
</tr>
<tr>
<td>JDK</td>
<td>Oracle JDK 1.6.0 or 1.7.0</td>
</tr>
</tbody>
</table>

**Configuring the JDK**

If you have not installed the Oracle JDK yet, download the file from the [http://www.oracle.com/technetwork/java/index.html](http://www.oracle.com/technetwork/java/index.html) website. Be sure to install the JDK, not the JRE.

The download site displays a number of software options. These might include terms such as JRE, JDK, and Java SDK. You must install a development kit (which is termed JDK or SDK) rather than installing only the JRE.

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.

After you install the Oracle JDK, you must configure it.

**To configure the JDK**

1. On your Windows machine, access the environment variables using the steps below:

   a. From the **Start** menu, select **Computer**. The Computer dialog box opens.
   b. Click **System properties**. The System dialog box opens.
   c. Click **Advanced system settings**. The System Properties dialog box opens.
   d. Click **Environment Variables**. The Environment Variables dialog box opens.

2. Under **System Variables**, click **New** to create a system variable. The New System Variable dialog box opens.

3. In the **Variable Name** field, type `JAVA_HOME`. 

   You can download the JDK here.
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.

After you install Tomcat, you must configure it.

---

**To configure Tomcat**

1 On your Windows machine, access the environment variables using the steps below:

   a From the **Start** menu, select **Computer**. The Computer dialog box opens.
   b Click **System properties**. The System dialog box opens.
   c Click **Advanced system settings**. The System Properties dialog box opens.
   d Click **Environment Variables**. The Environment Variables dialog box opens.

2 Under **System Variables**, click **New** to create a system variable. The New System Variable dialog box opens.

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`.
If you installed Tomcat under the Program Files folder, type Progra~1 when specifying the folder in the Variable Value box. Otherwise, the system does not recognize the folder. For example, type C:\Progra~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, Mobile Server, or Operations Manager

Assuming you have made all the necessary configurations described above, you can begin deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP) with Tomcat. This involves the following steps:

1. Deploying using Tomcat as a stand-alone Web container, page 233
2. Configuring administrative access your MicroStrategy JSP applications, page 235
3. Accessing the MicroStrategy JSP application administrative page, page 235
4. Launching the project, page 236

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 195.
2. Copy the WAR file to the Tomcat\webapps folder.

Stop and start Tomcat from the command line

3. From the Start menu, select Run. The Run dialog box opens.
4. Type cmd in the Open drop-down list and click OK. The command prompt opens.
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

Press ENTER.

C:\Tomcat\bin> is displayed at the command prompt.

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.
  - When deploying MicroStrategy Operations Manager (JSP), the folder is named MicroStrategyOM 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.
  - When deploying MicroStrategy Operations Manager (JSP), a /microstrategy/ipa-\Version/ folder is created within the temporary file directory, where Version is the version number for the MicroStrategy Operations Manager (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, or Operations Manager, 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 provided 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.

Now you can access and configure your MicroStrategy JSP application, as described in Accessing the MicroStrategy JSP application administrative page, page 235.

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 MicroStrategy JSP application administrative page

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
   - For Operations Manager (JSP):
     http://localhost:8080/MicroStrategyOM/servlet/mstrOM

   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 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 Design and Administration Guide**. Creating a configuration completes the steps required to deploy Mobile Server (JSP).
   • If you are deploying MicroStrategy Operations Manager (JSP), you can begin to monitor and manage your MicroStrategy server systems. For steps to create an environment, see the **Operations Manager Guide**.

4 If you are deploying MicroStrategy Web (JSP), proceed to launch the MicroStrategy project. For more information, see **Launching the project, page 236**.

**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 provided below to deploy MicroStrategy Web (JSP), MicroStrategy Mobile Server (JSP), or MicroStrategy Operations Manager (JSP):

- **Preconfiguration information, page 237**: Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP).
- **Deploying MicroStrategy Web, Mobile Server, or Operations Manager, page 239**: 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, page 237
- Configuring the JDK, page 238
- Installing Tomcat, page 238
- Configuring Tomcat, page 238
- Locating the WAR file, page 195

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>Red Hat Enterprise Linux 5.8 (on x86-64)</td>
</tr>
<tr>
<td>Application server</td>
<td>Tomcat 7.0.x or 8.0.x</td>
</tr>
<tr>
<td>JDK</td>
<td>Oracle JDK 1.6.0 or 1.7.0</td>
</tr>
</tbody>
</table>

You can download the JDK here.

Installing the JDK

If you have not installed Oracle JDK yet, download the shell file from the [http://www.oracle.com/technetwork/java/index.html](http://www.oracle.com/technetwork/java/index.html) website. Be sure to install the JDK, not the JRE.

The download site displays a number of software options. These might include terms such as JRE, JDK, and Java SDK. You must install a development kit (which is termed JDK or SDK) rather than installing only the JRE.

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
```
Confusing the JDK

After you install the Oracle JDK, you must configure it.

To configure the JDK

1. Open the /etc/profile file using a program that allows you to edit the file.
2. Add the following line:
   ```
   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 website

- Tomcat 7.0.x: website. [http://tomcat.apache.org/download-70.cgi](http://tomcat.apache.org/download-70.cgi)

1. 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.

To configure Tomcat

1. Open the etc/profile file in a program that allows you to edit the file.
2. Type the following:
   ```
   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 \texttt{JAVA\_OPTS} parameter in the \texttt{catalina.sh} file. For example, you can define this parameter as follows:

\begin{verbatim}
JAVA\_OPTS = "-Xms1024m -Xmx2048m"
\end{verbatim}

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, Mobile Server, or Operations Manager

After you have performed the configurations described above, you can begin deploying MicroStrategy JSP applications with Tomcat. This involves the following steps:

1. Deploying using Tomcat as a standalone Web container, page 239
2. Accessing the MicroStrategy JSP application administrative page, page 241

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 195.

2. Copy the WAR file to the \texttt{Tomcat/webapps} directory.

To start and stop Tomcat from the command line

3. Type \# \texttt{$CATALINA\_HOME/bin/startup.sh} and press 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 \texttt{Tomcat/webapps} folder:
     - When deploying MicroStrategy Web (JSP), the folder is named \texttt{MicroStrategy} by default.
     - When deploying MicroStrategy Mobile Server (JSP), the folder is named \texttt{MicroStrategyMobile} by default.
     - When deploying MicroStrategy Operations Manager (JSP), the folder is named \texttt{MicroStrategyOM} 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 UNIX and 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.
  ▪ When deploying MicroStrategy Operations Manager (JSP), a /microstrategy/ipa-Version/ folder is created within the temporary file directory, where Version is the version number for the MicroStrategy Operations Manager (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, or Operations Manager, 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 provided below.

**To configure administrative access to MicroStrategy JSP applications**

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:

   ```xml
   <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 241.
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://localhost:8080/MicroStrategy/servlet/mstrWebAdmin
   - For Mobile Server (JSP):
     http://localhost:8080/MicroStrategyMobile/servlet/mstrWebAdmin
   - For Operations Manager (JSP):
     http://localhost:8080/MicroStrategyOM/servlet/mstrOM

   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 Design and Administration Guide. Creating a configuration completes the steps required to deploy Mobile Server (JSP).
   - If you are deploying MicroStrategy Operations Manager (JSP), you can begin to monitor and manage your MicroStrategy server systems. For steps to create an environment, see the Operations Manager Guide.

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
steps provided below to deploy MicroStrategy Web (JSP), MicroStrategy Mobile Server (JSP), or MicroStrategy Operations Manager (JSP):

- **Preconfiguration information, page 242**: Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP).

- **Deploying MicroStrategy Web, Mobile Server, or Operations Manager, page 242**: Instructions for deploying the application.

### Preconfiguration information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on your machine. It also provides additional setup information for the machine where the application server and web server are already installed.

The configuration required to deploy MicroStrategy JSP applications on SAP is provided in the following table. While your configuration 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>Windows 2008 Enterprise Edition SP2 (on x64)</td>
</tr>
<tr>
<td>Application server</td>
<td>SAP NetWeaver Application Server 7.4.x</td>
</tr>
</tbody>
</table>

For information on the version numbers certified and supported by MicroStrategy, see the *MicroStrategy Readme*.

### Deploying MicroStrategy Web, Mobile Server, or Operations Manager

Once your machine has the necessary settings configured, you can deploy MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP) on the SAP-Windows machine. Deployment involves the following steps:

1. **Deploying MicroStrategy JSP applications with the SAP NetWeaver Application Server, page 242**

2. **Configuring administrative access to MicroStrategy JSP applications, page 244**

3. **Accessing the MicroStrategy JSP applications, page 244**

### 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 195*.

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**. The Run dialog box opens.

4. In the **Open** drop-down list, type `cmd`, and click **OK**. A command prompt opens.

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. From the **Operation Management** tab, select the **Systems** tab, and then click **Start & Stop**.

10. Select **Java EE Applications**. A list of applications deployed on the application server are displayed.

11. Select the MicroStrategy JSP application just deployed from the list.

12. From the **Application Details** section, select the **Status** tab, and then click **Start**.

13. Select **On all instances and Set "Started" as Initial State**.

   When application is started, the status is displayed as Started.
Configuring administrative access to MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, or Operations Manager, 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 244.

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
   - For Operations Manager (JSP):
     http://MachineName:PortNumber/MicroStrategyOM/servlet/mstrOM

   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 Design and Administration Guide. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

- If you are deploying MicroStrategy Operations Manager (JSP), you can begin to monitor and manage your MicroStrategy server systems. For steps to create an environment, see the Operations Manager Guide.

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 steps provided below to deploy MicroStrategy Web (JSP), MicroStrategy Mobile Server (JSP), or MicroStrategy Operations Manager (JSP):

- Preconfiguration information, page 245: Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP).

- Deploying MicroStrategy Web, Mobile Server, or Operations Manager, page 246: Instructions for deploying, including detailed steps.

Preconfiguration information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on your machine.

This section uses the following setup. 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>Windows 2008 SP2 (on x64)</td>
</tr>
<tr>
<td>Application server</td>
<td>Oracle Application Server 10g 10.1.3</td>
</tr>
<tr>
<td>Web server</td>
<td>Apache 2.x</td>
</tr>
<tr>
<td>JDK</td>
<td>Oracle JDK 1.6.0 or 1.7.0</td>
</tr>
</tbody>
</table>

You can download the JDK here.
Deploying MicroStrategy Web, Mobile Server, or Operations Manager

After your machine is configured, you can start the deployment of your MicroStrategy JSP application with Oracle Application Server 10g R3.

To deploy MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP) perform the following procedures:

1. *Deploying using the Oracle Enterprise Manager, page 246*
2. *Accessing the MicroStrategy JSP application administrative page, page 241*

**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.

---

**To deploy with the Oracle Enterprise Manager**

1. Start the Apache Web Server. From the Start menu, point to OracleAS 10g - DEFAULT_HOME1, and then choose 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. The Deploy: Select Archive page opens.
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 195*.
8. In the Deployment Plan area, select Automatically create a new deployment plan and click Next. The Deploy: Application Attributes page opens.
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. For Operations Manager (JSP), this section uses MicroStrategyOM as the Application Name and /MicroStrategyOM as the Context Root.

10 Click Next. The Deploy: Deployment Settings page opens.

**To map a user to the admin security role**

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, or Operations Manager, you must assign users the security role of admin. In Oracle 10g, the security users and groups are defined in the Oracle Enterprise Manager.

11 In the Map Security Roles task name, click the Go To Task (pencil) icon. The Deployment Settings: Map Security Roles page opens.

12 For the admin security role, select the Map Role (pencil) icon. The Deployment Settings: Map Security Role: admin page opens.

13 Select **Map selected users and groups to this role**.

14 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.

15 Click Continue, and then click OK. You are returned to the Deploy: Deployment Settings page.

16 Click on Deploy to deploy the application.

17 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 247.

**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://
**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 provided below to deploy MicroStrategy Web (JSP), MicroStrategy Mobile Server (JSP), or MicroStrategy Operations Manager (JSP):

- **Preconfiguration information, page 249:** configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP).

### Preconfiguration information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on JBoss on your machine.

This section uses the following configuration. 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>Windows 2008 Enterprise Edition SP2 (on x64)</td>
</tr>
<tr>
<td>Application server</td>
<td>JBoss 7.4.x</td>
</tr>
<tr>
<td>JDK</td>
<td>Oracle JDK 1.6.0 or 1.7.0</td>
</tr>
</tbody>
</table>

| You can download the JDK here. |

### Configuring the JDK

If you have not installed Oracle JDK yet, download them from the [http://www.oracle.com/technetwork/java/index.html](http://www.oracle.com/technetwork/java/index.html) website.

When you go to the download site, you may be presented with a number of software options. These might include terms such as JRE, JDK, and Java SDK. You must install a developer kit (which can be termed JDK or SDK) rather than installing only the JRE.

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.

After you install the Oracle JDK, you must configure it.

### To configure the JDK

1. On your Windows machine, access the environment variables using the steps below:

   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.

   a. From the **Start** menu, select **Computer**. The Computer dialog box opens.

   b. Click **System properties**. The System dialog box opens.
c Click Advanced system settings. The System Properties dialog box opens.
d Click Environment Variables. The Environment Variables dialog box opens.

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:\jsdk1.6.0\bin\java.exe, the value of your JAVA_HOME variable is C:\jsdk1.6.0.

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\jsdk1.6.0.

**Installing JBoss**

You can download and install JBoss from the following website:


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, Mobile Server, or Operations Manager**

Assuming you have made all the necessary configurations described above, you can begin deploying MicroStrategy Web (JSP), Mobile Server (JSP), or Operations Manager (JSP) with JBoss. This involves the following steps:

1. **Deploying using JBoss as a stand-alone Web container, page 251**

2. **Configuring administrative access to MicroStrategy JSP applications, page 252**

3. **Accessing the MicroStrategy JSP application administrative page, page 253**
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 195.

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.
  - When deploying MicroStrategy Operations Manager (JSP), the folder is named MicroStrategyOM 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 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.
  - When deploying MicroStrategy Operations Manager (JSP), a /microstrategy/ipa-Version/ folder is created within the temporary file directory, where Version is the version number for the MicroStrategy Operations Manager (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, or Operations Manager, you must create the users and assign them the role of admin under the JBoss user configuration files. The steps to configure this access are provided below.

To configure administrative access to MicroStrategy JSP applications

1. Browse to the directory JBOSS_HOME\server\default\conf, where 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, Mobile Server Administrator, and Operations Manager, 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 253.
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):
     http://localhost:8080/MicroStrategy/servlet/mstrWebAdmin
   - For Mobile Server (JSP):
     http://localhost:8080/MicroStrategyMobile/servlet/mstrWebAdmin
   - For Operations Manager (JSP):
     http://localhost:8080/MicroStrategyOM/servlet/mstrOM

   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 Design and Administration Guide**. Creating a configuration completes the steps required to deploy Mobile Server (JSP).
   - If you are deploying MicroStrategy Operations Manager (JSP), you can begin to monitor and manage your MicroStrategy server systems. For steps to create an environment, see the **Operations Manager Guide**.

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 configure and manage MicroStrategy Web connections to Intelligence Servers in the MicroStrategy Web Administrator page.

**Enabling users to install MicroStrategy Office from Web**

From the MicroStrategy Web Administrator page, you can designate the installation directory path to MicroStrategy Office, and also determine whether a link to Office installation information appears in the MicroStrategy Web interface.

**Prerequisites**


---

**To specify the path to MicroStrategy Office and determine whether users can install MicroStrategy Office from Web**

1. From the Windows **Start** menu, select **Programs, MicroStrategy Tools**, and then **Web Administrator**. The MicroStrategy Web Administrator page opens.

   If your server is not connected, click **Connect**.


3. 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 /Langxxxx/officeinstall.htm to the end of the URL, where Lang_xxxx 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
   ```

4. Test the URL path by clicking **Go**. If the path you specified is correct, the MicroStrategy Office Installation page is displayed.
5 Click your browser's Back button to return to the Web Administration - MicroStrategy Office settings page.

6 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.

7 Click Save to save the settings.

Using absolute paths to share configuration files

By default, absolute paths are not used for the configuration files of your MicroStrategy Web, Mobile, or Operations Manager 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
The steps below show you how to make these third-party data sources available for import into MicroStrategy Web.

**Prerequisites**

- 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 provided 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 402](#).

**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/](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:
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.

For steps to use Data Import to integrate data into MicroStrategy, refer to the MicroStrategy Web Help.

### Configuring your MicroStrategy installation

After completing the steps to deploy MicroStrategy Web, Mobile Server, or Operations Manager, 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 48* in Chapter 1, Planning Your Installation, for installation and configuration checklists.
7

Setting Up Documents and HTML Documents

This chapter explains the setup required for Intelligence Server to execute HTML
documents and Report Services documents on UNIX and Linux platforms.

This chapter includes the following sections:

- **Prerequisites** ................................................................. 258
- **Executing documents and HTML documents in UNIX/Linux** .................. 259
- **Configuring your MicroStrategy installation** .................................. 264

Prerequisites

This chapter assumes the following:

- You are familiar with UNIX or Linux.
- 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 UNIX or Linux machine.
Note the following:

- 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 434 of Appendix C, Troubleshooting.*

## Executing documents and HTML documents in UNIX/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 chapter, 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 dashboards 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 UNIX or 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 UNIX or Linux platform, see *Setup for creating and executing HTML documents, page 259.*

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 UNIX or 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 UNIX 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 UNIX or Linux computer.

  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 UNIXi, `machine-name` should be replaced with UNIXi. This machine must have Samba installed to provide access to the folder on a Windows computer.

- You must have root permissions on any UNIX or 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 UNIX or 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 UNIX or 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 a .org extension and not the more common .com extension. Using a .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 UNIX/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 `/htmldocuments` and `/machine-name`.

```bash
cd /
mkdir machine-name
cd machine-name
mkdir share
cd share
mkdir htmldocuments
cd /
mount machine-name:/share/htmldocuments /machine-name/share/htmldocuments
```

7 You can cache the connection to the UNIX/Linux HTML documents directory from the Windows computer so that you are not prompted for authentication each time the directory is accessed:
From the Windows computer that has MicroStrategy Developer installed, click **Start**, and select **Run**. The Run dialog box opens.

Type `\machine-name\share\htmldocuments`, and click **OK** to open the top-level shared HTML documents directory. For example, type `\UNIX1\share\htmldocuments`.

This must be performed every time you restart the computer.

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.

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.

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 UNIX or 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 259*. 

© 2017, MicroStrategy Inc.
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="\xsls\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 UNIX or Linux environment, you must perform some additional setup tasks.

When Intelligence Server is running on a UNIX or 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 UNIX and 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 UNIX or Linux machine, as described below.

**To copy fonts to your UNIX or Linux machine**

1. Log in to your UNIX or 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 UNIX and 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 48* in Chapter 1, *Planning Your Installation*, for installation and configuration checklists.
Automated Installation on Windows

This chapter 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 chapter provides the following information:

- The different types of installations that can be performed with MicroStrategy products.
- How to perform a fully automated MicroStrategy installation by modifying various installation parameters in Windows ini-like response files.
- How to customize certain aspects of the MicroStrategy installation to meet various site-specific requirements for multi-user environments, such as strict standards for software deployment to user communities and so on.

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 chapter are:

- *Installing and configuring with a response.ini file, page 267*
- *Silent installation, page 315*
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, page 267
- Configuring your installation with a response.ini file, page 313
- Uninstalling with a response.ini file, page 313

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:

- 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 315.

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.
require MicroStrategy Narrowcast Administrator and SequeLink ODBC Socket Server 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 Office can be installed along with other MicroStrategy components by using a response.ini file. However, you can also install MicroStrategy Office as its own stand-alone installation, which lets you install only MicroStrategy Office. For information on using the stand-alone installation of MicroStrategy Office, including performing the installation silently, see Silent installation of MicroStrategy Office, page 318.

- To use Usher Security Server and Usher Network Manager, they are required to be included in the installation.

- Usher Analytics requires Usher Security Server and Usher Network Manager to be included in the installation.

- Usher Professional requires Usher Security Server, Usher Network Manager and Usher Analytics 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>ExpressMode =</td>
<td>TRUE or FALSE. Indicates whether the installation uses Express Install or Custom Install.</td>
</tr>
<tr>
<td>HideAllDialogs =</td>
<td>TRUE or FALSE. 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>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>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\Common Files\MicroStrategy\install.log.</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\Common Files\MicroStrategy\install.log.</td>
</tr>
<tr>
<td>CreateShortcuts =</td>
<td>TRUE or FALSE. TRUE creates the shortcuts for MicroStrategy products, tools, and documentation. The default is TRUE.</td>
</tr>
<tr>
<td>PreventReboot =</td>
<td>TRUE or FALSE. TRUE prevents the machine from rebooting after installation is done. Note the following conditions:</td>
</tr>
<tr>
<td></td>
<td>• If both ForceReboot = TRUE and PreventReboot = TRUE, then PreventReboot applies first.</td>
</tr>
<tr>
<td></td>
<td>• If both ForceReboot and PreventReboot are FALSE and HideDialog for [Finish] is set to TRUE, then the machine reboots only if it is required.</td>
</tr>
<tr>
<td></td>
<td>The default is FALSE.</td>
</tr>
<tr>
<td>ForceReboot =</td>
<td>TRUE or FALSE. TRUE reboots the machine after the installation is done. The default is FALSE.</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>ShowConfigWizard =</td>
<td>TRUE or FALSE. When using silent install, set to FALSE to prevent the Configuration Wizard from coming up after reboot. The default is TRUE.</td>
</tr>
<tr>
<td>RunConfigWizard =</td>
<td>TRUE or FALSE. Set this option to specify whether to run the Configuration Wizard. The default is FALSE.</td>
</tr>
<tr>
<td>ConfigWizardResponse</td>
<td>Specify the name of the response file for the Configuration Wizard;</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>File</td>
<td>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 313.</td>
</tr>
</tbody>
</table>
| BackupFiles     | TRUE or FALSE. If you set the value to TRUE, it creates a backup of the following files:  
  - *.pds  
  - *.xsl  
  - *.asp  
  - *.css  
  - *.js  
  - *.sql  
  The default is FALSE. |
| StopIIS         | TRUE or FALSE. Set this option to stop Internet Information Services (IIS) during installation. The default is FALSE.                           |
| AnalyticsOverwrite | TRUE or FALSE. This option overwrites the Analytics Modules from a previous install. The default is FALSE.                                       |
| TutDeliveryOverwrite | TRUE or FALSE. Set this option to overwrite the Delivery Tutorial from a previous installation. The default is FALSE.                           |
| CheckRenameOperations | TRUE or FALSE. 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:  
  - Reboot at the beginning of installation to replace this file. It is recommended that you select this option.  
  - Continue with the installation at the risk of the software not functioning properly.  
  If you enter FALSE, the prompt does not display. The default is TRUE. |
| EnableTracing   | TRUE or FALSE. 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. |
| EnableASPNETServices | TRUE or FALSE. Set this option to enable the ASP.NET MicroStrategy Web Services extensions that IIS Admin requires. The default is FALSE. |
Options | Description
--- | ---
EnableASPServices = TRUE or FALSE. Set this option to enable the ASP MicroStrategy Web Services extensions that IIS Admin requires. The default is FALSE.
ShowWelcomeScreen = 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.
PropertiesFilesOverwrite = TRUE or FALSE. 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.
CheckTCPIP = TRUE or FALSE. 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.
CheckIIS = TRUE or FALSE. Set to TRUE to check for Internet Information Services. The default is TRUE.
StopAllServices = 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.

**Setup Express Install**

Options | Description
--- | ---
[SetupExpress] | Section that begins the installation.
HideDialog = TRUE or False. Indicates whether the installation uses Express Install or Custom Install.

**Paths**

Options | Description
--- | ---
[InitialPaths] | Section for specifying the path for the products that you select to install.
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\Common Files\MicroStrategy</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 (x86)\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 (x86)\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 (x86)\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 (x86)\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></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Intelligence Server</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Web =</strong></td>
<td>Location where MicroStrategy Web will be installed. If left empty, it takes</td>
</tr>
<tr>
<td></td>
<td>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><strong>WebUniversal =</strong></td>
<td>Location where MicroStrategy Web will be installed. If left empty, it takes</td>
</tr>
<tr>
<td></td>
<td>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><strong>WebServices =</strong></td>
<td>Location where MicroStrategy Web Services will be installed. If left empty,</td>
</tr>
<tr>
<td></td>
<td>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><strong>WebServicesUniversal =</strong></td>
<td>Location where MicroStrategy Web Services will be installed. If left empty,</td>
</tr>
<tr>
<td></td>
<td>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><strong>Office =</strong></td>
<td>Location where MicroStrategy Office will be installed. If left empty, it</td>
</tr>
<tr>
<td></td>
<td>takes the default location of:</td>
</tr>
<tr>
<td></td>
<td>• 32-bit Windows environments: C:\Program Files\MicroStrategy\Office</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Office</td>
</tr>
<tr>
<td><strong>TutorialReporting =</strong></td>
<td>Location where MicroStrategy Tutorial - Reporting will be installed. If left</td>
</tr>
<tr>
<td></td>
<td>empty, it takes the default location of:</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>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Analytics Modules =</td>
<td>Location where the Analytics Modules 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\Analytics Modules</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Analytics Modules</td>
</tr>
<tr>
<td>NCSAdminDelivery Engine =</td>
<td>Location where the Narrowcast Server Delivery Engine 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\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>Subscription Portal =</td>
<td>Location where the Subscription Portal 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\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>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>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</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>MobileJSSPath =</td>
<td>Location where MicroStrategy Mobile Server JSP 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 JSP</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Mobile Server JSP</td>
</tr>
<tr>
<td>OperationsManagerASP =</td>
<td>Location where MicroStrategy Operations Manager 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\Operations Manager ASPx</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Operations Manager ASPx</td>
</tr>
<tr>
<td>OperationsManagerJSP =</td>
<td>Location where MicroStrategy Operations Manager JSP 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\Operations Manager JSP</td>
</tr>
<tr>
<td></td>
<td>• 64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Operations Manager JSP</td>
</tr>
<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 (x86)\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>
</tbody>
</table>
Options | Description
--- | ---
GISConnectors = | Location where the MicroStrategy GIS Connectors will be installed. If left empty, it takes the default location of:
- **32-bit Windows environments:** `C:\Program Files\MicroStrategy\GISConnectors`
- **64-bit Windows environments:** `C:\Program Files (x86)\MicroStrategy\GISConnectors`

SystemManager = | Location where the MicroStrategy System Manager will be installed. If left empty, it takes the default location of:
- **32-bit Windows environments:** `C:\Program Files\MicroStrategy\SystemManager`
- **64-bit Windows environments:** `C:\Program Files (x86)\MicroStrategy\SystemManager`

---

**Welcome dialog box**

Options | Description
--- | ---
[Welcome] | Section for configuring the Welcome dialog box.
HideDialog = | TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
RemoveAll = | 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.

For an example of a response file used to uninstall all MicroStrategy products, see *Uninstalling with a response.ini file, page 313.*

---

**Customer Information dialog box**

Options | Description
--- | ---
[UserRegistration] | Section for specifying the customer information.
HideDialog = | TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
UserFirstName = | 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 Usher, this is the first name of the
### Options

<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 Usher, this is the last name of the Usher Security 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 Usher Security 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 Usher Security 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>

### Choose Destination Location dialog box

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</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>

### Select Components dialog box

In the MicroStrategy Installation Wizard, the Select Components dialog box contains check boxes 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</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HideDialog = TRUE or FALSE. FALSE displays</td>
<td>the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>the dialog box. The default is FALSE.</td>
<td></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>
<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>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>NCSAdminVisible =</td>
<td>MicroStrategy Narrowcast Administrator</td>
</tr>
<tr>
<td>ObjectManagerVisible =</td>
<td>MicroStrategy Object Manager</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>OfficeVisible =</td>
<td>MicroStrategy Office</td>
</tr>
<tr>
<td>OperationsManagerASPVisible =</td>
<td>MicroStrategy Operations Manager (ASP.NET)</td>
</tr>
<tr>
<td>OperationsManagerJSPVisible =</td>
<td>MicroStrategy Operations Manager (JSP)</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>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>Usher Security Server</td>
</tr>
<tr>
<td>UsherNetworkManagerVisible =</td>
<td>Usher Network Manager</td>
</tr>
<tr>
<td>UsherAnalyticsVisible =</td>
<td>Usher Analytics</td>
</tr>
<tr>
<td>UsherProfessionalVisible =</td>
<td>Usher Professional</td>
</tr>
</tbody>
</table>

During the installation process in the MicroStrategy Installation Wizard, the Select Components dialog box contains check boxes 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 check box is selected. The [ComponentSelection] options specify whether the check box for each product will be selected or cleared.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>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>MobileSelect =</td>
<td>MicroStrategy Mobile</td>
</tr>
<tr>
<td>MobileServerASPSelect =</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>OperationsManagerASPSelect =</td>
<td>MicroStrategy Operations Manager (ASP.NET)</td>
</tr>
<tr>
<td>OperationsManagerJSPSelect =</td>
<td>MicroStrategy Operations Manager (JSP)</td>
</tr>
<tr>
<td>PortletsSelect =</td>
<td>MicroStrategy Portlets</td>
</tr>
<tr>
<td>SequeLinkSelect =</td>
<td>SequeLink ODBC Socket Server</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>ServerAdminSelect</code> =</td>
<td>MicroStrategy Server Administrator</td>
</tr>
<tr>
<td><code>SubscriptionPortalSelect</code> =</td>
<td>MicroStrategy Subscription Portal</td>
</tr>
<tr>
<td><code>SystemManagerSelect</code> =</td>
<td>MicroStrategy System Manager</td>
</tr>
<tr>
<td><code>MDXCubeProviderSelect</code> =</td>
<td>MicroStrategy MDX Cube Provider</td>
</tr>
<tr>
<td><code>TutorialDeliveryConfigureSelect</code> =</td>
<td>MicroStrategy Tutorial - Delivery Configuration</td>
</tr>
<tr>
<td><code>TutorialDeliveryInstallSelect</code> =</td>
<td>MicroStrategy Tutorial - Delivery Installation</td>
</tr>
<tr>
<td><code>WebAnalystSelect</code> =</td>
<td>MicroStrategy Web Analyst</td>
</tr>
<tr>
<td><code>WebProfessionalSelect</code> =</td>
<td>MicroStrategy Web Professional</td>
</tr>
<tr>
<td><code>WebReporterSelect</code> =</td>
<td>MicroStrategy Web Reporter</td>
</tr>
<tr>
<td><code>WebServerASPNETSelect</code> =</td>
<td>MicroStrategy Web Server (ASP.NET)</td>
</tr>
<tr>
<td><code>WebServerJSPSelect</code> =</td>
<td>MicroStrategy Web Server JSP</td>
</tr>
<tr>
<td><code>WebServicesASPNETSelect</code> =</td>
<td>MicroStrategy Web Services (ASP.NET)</td>
</tr>
<tr>
<td><code>WebServicesJSPSelect</code> =</td>
<td>MicroStrategy Web Services (JSP)</td>
</tr>
<tr>
<td><code>UsherServerVisible</code> =</td>
<td>Usher Security Server</td>
</tr>
<tr>
<td><code>UsherNetworkManagerVisible</code> =</td>
<td>Usher Network Manager</td>
</tr>
<tr>
<td><code>UsherAnalyticsVisible</code> =</td>
<td>Usher Analytics</td>
</tr>
<tr>
<td><code>UsherProfessionalVisible</code> =</td>
<td>Usher Professional</td>
</tr>
</tbody>
</table>

### Installation Files

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>[IODSourceLocation]</code></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 46.</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Style =</strong></td>
<td>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:</td>
</tr>
<tr>
<td><code>FILESERVER</code></td>
<td>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 <code>SourceLocation</code> parameter.</td>
</tr>
<tr>
<td><code>HTTP</code></td>
<td>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 <code>URL</code> parameter.</td>
</tr>
<tr>
<td><code>HTTPS</code></td>
<td>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 <code>URL</code> parameter, as well as the user name and password to access the URL using the <code>UserName</code> and <code>Password</code> parameters.</td>
</tr>
</tbody>
</table>

| SourceLocation = | 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. |

| URL = | 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. |

| UserName = | 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. |

| Password = | 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. |

### Usher Configuration

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>[UsherConfig]</code></td>
<td>Section that configures Usher Security Services.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td><code>TRUE</code> or <code>FALSE</code>. <code>FALSE</code> displays the dialog box. The default is <code>FALSE</code>.</td>
</tr>
<tr>
<td>ExpressSkipUsherConfig=</td>
<td><code>TRUE</code> or <code>FALSE</code> when <code>ExpressMode= TRUE</code>. <code>TRUE</code> to skip Usher configuration in Express Installation. Set to <code>FALSE</code> when <code>ExpressMode= FALSE</code>.</td>
</tr>
</tbody>
</table>
**Options** | **Description**  
--- | ---  
CACertificateChain = | 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.  
ServerCertificate = | 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.  
ServerCertificateKey = | The key file for your SSL Server Certificate (.crt). The path must be specified in an absolute format such as C:\folder\example.key.  
ServerCertificateKeyPasswordFile = | If your CA-signed certificate has a password, create a text file containing this password and enter the text file location.  
SMTPServer = | SMTP Server used for Usher email service.  
SMTPServerPort = | SMTP Server Port.  
SMTPUser = | If your server is password protected, enter the username for the server. This is optional.  
SMTPUserPassword = | If your server is password protected, enter the password for the server. This is optional.  
SMTPEmail = | The email address that is authorized to send emails from your SMTP server. This email address is used to send badge invitations for your Usher Security Network.  
FQDN = | The Fully Qualified Domain Name of your Windows server.  

**Repository Configuration**

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
</table>
| [MySQLDialog] | MySQL is required as a repository in Express Install or required by Usher products in Custom Install.  
HideDialog = | TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.  
AgreeToDownloadMySQL= | TRUE or FALSE. TRUE to allow the installed to download the MySQL components for you. FALSE to follow the links below to download the MySQL 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 with Usher products cannot proceed.  

Options | Description
--- | ---
|  | Please note that if you choose TRUE, you are authorizing the installer to download MySQL components on your behalf. These MySQL components are open-source software provided under the GPL licenses. These MySQL components are not provided by MicroStrategy. For access to the source code for these MySQL components, please visit [http://www.mysql.com](http://www.mysql.com) and related links.
| MySQL: | [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) |
| MySQL Connector/ODBC 5.3.4: | [http://dev.mysql.com/get/Downloads/Connector-ODBC/5.3/mysql-connector-odbc-5.3.4-winx64.msi](http://dev.mysql.com/get/Downloads/Connector-ODBC/5.3/mysql-connector-odbc-5.3.4-winx64.msi) |
| MySQL Connector/Python 2.1.3 | [http://dev.mysql.com/get/Downloads/Connector-Python/mysql-connector-python-2.1.3-py2.7-winx64.msi](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](http://downloads.mysql.com/general/timezone_2015g_posix.zip) |

### 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 this option to remove an existing MicroStrategy Web virtual directory from a previous installation. The default is NO.</td>
</tr>
</tbody>
</table>
### MicroStrategy Mobile Server virtual directory

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>

### Health Center service account

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[HealthCenterServiceAccount]</td>
<td>Section that configures the machine running the silent installation as a Health Agent or Master Health Agent.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>Port =</td>
<td>Type the port number to use to connect to the Health Agent machine. The default port is 44440.</td>
</tr>
<tr>
<td>AccessCode =</td>
<td>Type the access code that must be provided by Health Center to access this Health Agent. If you leave this field blank, no access code is required to access this Health Agent.</td>
</tr>
<tr>
<td>SkipAccountSetting =</td>
<td>TRUE or FALSE. FALSE requires that the machine is configured to be a Health Agent, and you must supply the other information. The default is TRUE, which skips the configuration of the machine as a Health Agent.</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain =</td>
<td>Type the domain of a Windows login with full administrative privileges under which to run the Health Center service.</td>
</tr>
</tbody>
</table>
| Login =                     | Type the user name of a Windows login with full administrative privileges under which to run the Health Center service.  
|                             | The user account used to run Health Center 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. |
| Password =                  | Type a valid password for the Windows login provided.                       |
| MasterHealthAgent =         | TRUE or FALSE. TRUE specifies the current machine as a Master Health Agent, which is responsible for most of the Health Center operations, such as scheduling system checks and transmitting diagnostics packages to MicroStrategy Technical Support. If you install MicroStrategy Operations Manager, the default is TRUE, otherwise the default is FALSE. |
| RepositoryPath =            | Type the location to store the Health Center repository. The repository contains configuration information about the Health Center system, such as the list of machines on the network and the MicroStrategy products they have installed, and also the destination for all exported diagnostics packages. |
| CustomerExperienceProgram = | TRUE or FALSE. TRUE enrolls the installation in the Customer Experience Improvement Program. If you join the Customer Experience Improvement Program, Health Center transmits anonymous data about your system to MicroStrategy. No report data or prompt answers are collected or transmitted. All information sent to MicroStrategy as a result of this program is stored in the Census subfolder of the Health Center Repository. The default is FALSE. |

---

**MicroStrategy Operations Manager virtual directory**

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[OperationsManagerVirtualDirectory]</td>
<td>Section that specifies the virtual directory to be</td>
</tr>
</tbody>
</table>
## Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HideDialog = TRUE or FALSE</td>
<td>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 MicroStrategyOM.</td>
</tr>
<tr>
<td>ReconfigureVirtualDirectory =</td>
<td>TRUE or FALSE. Set this option to TRUE if the virtual directory for the MicroStrategy Operations Manager should be reconfigured to support a new virtual directory.</td>
</tr>
<tr>
<td>RemoveVD = YES or NO</td>
<td>YES or NO. This option is for the uninstall only. Set this option to remove an existing MicroStrategy Operations Manager 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 = TRUE or FALSE</td>
<td>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 = YES or NO</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>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</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>
<tr>
<td>RemoveVD =</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>
<tr>
<td></td>
<td><strong>MicroStrategy Web Services virtual directory</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>[WebServicesDirectory]</td>
<td>Section that specifies the virtual directory to be used for MicroStrategy Web Services.</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 MicroStrategyWS.</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 Subscription Portal virtual directory from a previous installation. The default is NO.</td>
</tr>
<tr>
<td></td>
<td><strong>Intelligence Server service account</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<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 =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>SkipAccountSetting =</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 the service runs with the local system account that is installing the products. The default is FALSE.</td>
</tr>
<tr>
<td>Domain =</td>
<td>Enter the domain where the account is located.</td>
</tr>
</tbody>
</table>
### Options Description

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
<tr>
<td>ServiceStartup = AUTO or MANUAL. Select to set the Intelligence Server service startup to be automatic or manual. The default is AUTO.</td>
<td></td>
</tr>
</tbody>
</table>

### Narrowcast Server service account

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[NarrowcastServiceAccount] Section specifying the Windows account from which the MicroStrategy Narrowcast Server service will run.</td>
<td></td>
</tr>
<tr>
<td>HideDialog = TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
<td></td>
</tr>
<tr>
<td>SkipAccountSetting = 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>
<td></td>
</tr>
<tr>
<td>Domain = Enter the domain where the account is located.</td>
<td></td>
</tr>
<tr>
<td>Login = Enter the user name of the account to use.</td>
<td></td>
</tr>
<tr>
<td>Password = Enter the password for the account.</td>
<td></td>
</tr>
</tbody>
</table>

### MicroStrategy Web Services and Web Services URL setting

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HideDialog = TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
<td></td>
</tr>
<tr>
<td>AllowBlankURL = 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>
<td></td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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

<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 Microsoft Office applications.</td>
</tr>
<tr>
<td>HideDialog =</td>
<td>TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.</td>
</tr>
<tr>
<td>ConfigureExcel =</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 =</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 =</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 definition setting

<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 setting

<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>

Start Copying Files dialog box

<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 box

<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.

[Installer]
ExpressMode=FALSE
PropertiesFilesOverwrite=FALSE
EnableTracing=FALSE
HideAllDialogs=TRUE
ForceReboot=TRUE
PreventReboot=FALSE
CheckTCPIP=TRUE
CheckIIS=TRUE
CreateShortcuts=TRUE
CheckRenameOperations=TRUE
AnalyticsOverwrite=TRUE
TutDeliveryOverwrite=TRUE
BackupFiles=TRUE
RunConfigWizard=FALSE
StopAllServices=TRUE
StopIIS=TRUE
EnableASPServices=TRUE
EnableASPNETServices=TRUE
ShowWelcomeScreen=FALSE
ShowConfigWizard=TRUE
EnterpriseManagerOverwrite=TRUE
ConfigWizardResponseFile=
LogFile=C:\Program Files (x86)\Common Files\MicroStrategy\install.log

[Welcome]
HideDialog=TRUE
RemoveAll=FALSE

[UserRegistration]
HideDialog=TRUE
UserFirstName=<UserFirstName>
UserLastName=<UserLastName>
UserEmail=<UserEmail>
CompanyName=<CompanyName>
LicenseKey=<CustomerLicenseKey>

[SetupExpress]
HideDialog=TRUE

[SuiteTarget]
HideDialog=TRUE
TargetDirectory=C:\Program Files (x86)\MicroStrategy

[ComponentSelection]
HideDialog=TRUE

### Visible Components ###
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
Web AnalystVisible=TRUE
Web ProfessionalVisible=TRUE
Web ReporterVisible=TRUE
Web Server ASP.NETVisible=TRUE
Web Server JSPVisible=TRUE
Web Services ASP.NETVisible=TRUE
Web Services JSPVisible=TRUE
OfficeVisible=TRUE
MobileVisible=TRUE
Mobile ClientVisible=TRUE
Mobile Server ASPVisible=TRUE
Mobile Server JSPVisible=TRUE
Analytics ModulesVisible=TRUE
NCS AdminVisible=TRUE
Delivery EngineVisible=TRUE
Subscription PortalVisible=TRUE
Tutorial Delivery InstallVisible=TRUE
Tutorial Delivery ConfigureVisible=TRUE
SequeLinkVisible=TRUE
PortletsVisible=TRUE
MDX Cube ProviderVisible=TRUE
GIS ConnectorsVisible=TRUE
System ManagerVisible=TRUE
Operations Manager ASPVisible=TRUE
Operations Manager JSPVisible=TRUE
Usher ServerVisible=TRUE
Usher Network ManagerVisible=TRUE
Usher AnalyticsVisible=TRUE
Usher ProfessionalVisible=TRUE
### Components To Install (TRUE) or Remove (FALSE) ###
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
NCAdminSelect=TRUE
DeliveryEngineSelect=TRUE
SubscriptionPortalSelect=TRUE
TutorialDeliveryInstallSelect=TRUE
TutorialDeliveryConfigureSelect=TRUE
SequeLinkSelect=TRUE
PortletsSelect=TRUE
MDXCubeProviderSelect=TRUE
GISConnectorsSelect=TRUE
SystemManagerSelect=TRUE
OperationsManagerASPSelect=TRUE
OperationsManagerJSPSelect=TRUE
UsherServerSelect=TRUE
UsherNetworkManagerSelect=TRUE
UsherAnalyticsSelect=TRUE
UsherProfessionalSelect=TRUE
[UsherConfig]
HideDialog=TRUE
ExpressSkipUsherConfig=FALSE
CACertificateChain=<AbsolutePath_CACertificateChainFile>
ServerCertificate=<AbsolutePath_SSLServerCertificateFile>
ServerCertificateKey=<AbsolutePath_SSLServerCertificateKeyFile>
ServerCertificateKeyPasswordFile=<Optional.AbsolutePath_ServerCertificateKeyPasswordFile>
SMTPServer=<SMTPServer>
SMTPServerPort=<SMTPServerPort>
SMTPUser=<Optional_SMTPUser>
SMTPUserPassword=<Optional_SMTPUserPassword>
SMTPEmail=<EmailSenderAddress>
FQDN=<FQDN>
[MySQLDialog]
HideDialog=TRUE
AgreeToDownloadMySql=TRUE
[ServerDefinitionSetting]
HideDialog=TRUE
OverwriteServerDefinition=FALSE
[HealthCenterServiceAccount]
HideDialog=TRUE
Port=<PortNumber>
AccessCode=<AccessCode>
SkipAccountSetting=FALSE
Domain=<WindowsLoginDomain>
Login=<WindowsLogin>
Password=<WindowsLoginPassword>
MasterHealthAgent=TRUE
RepositoryPath=C:\HealthCenterRepo
CustomerExperienceProgram=FALSE
[AnalyticsSetting]
HideDialog=TRUE
OverwriteDSN=FALSE
[WebVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategy
ReconfigureVirtualDirectory=TRUE
[MobileVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyMobile
ReconfigureVirtualDirectory=TRUE
[OperationsManagerVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyOM
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 IServerOLAPServicesSelect=TRUE and IServerOLAPServicesVisible=TRUE to install these components.

**Example of a response.ini file for Express Installation**

Starting from 10.4 you can find the sample_express.ini in the same location as the installation setup.exe. Replace any text between angled brackets <> with your own specific information.

```
[Installer]
ExpressMode=TRUE
PropertiesFilesOverwrite=
EnableTracing=
HideAllDialogs=TRUE
ForceReboot=TRUE
PreventReboot=
CheckTCPIP=
CheckIIS=TRUE
CreateShortcuts=
CheckRenameOperations=
AnalyticsOverwrite=
TutDeliveryOverwrite=
BackupFiles=
RunConfigWizard=FALSE
StopAllServices=TRUE
StopIIS=TRUE
EnableASPServices=
EnableASPNETServices=
```
ShowWelcomeScreen=FALSE
ShowConfigWizard=FALSE
EnterpriseManagerOverwrite=
ConfigWizardResponseFile=
LogFile=
[Welcome]
HideDialog=TRUE
RemoveAll=FALSE
[UserRegistration]
HideDialog=TRUE
UserFirstName=<UserFirstName>
UserLastName=<UserLastName>
UserEmail=<UserEmail>
CompanyName=<CompanyName>
LicenseKey=<CustomerLicenseKey>
[SetupExpress]
HideDialog=TRUE
[Installer]
HideAllDialogs=TRUE
ForceReboot=TRUE
StopAllServices=TRUE
StopIIS=TRUE
EnterpriseManagerOverwrite=TRUE

Example of a response.ini file: Installs Intelligence Server, Enterprise Manager, and other components

[Installer]
HideAllDialogs=TRUE
ForceReboot=TRUE
StopAllServices=TRUE
StopIIS=TRUE
EnterpriseManagerOverwrite=TRUE
[Welcome]
HideDialog=TRUE
RemoveAll=FALSE

[UserRegistration]
HideDialog=TRUE
UserName=<UserNameHere>
CompanyName=<CompanyNameHere>
LicenseKey=<CustomerLicenseKeyHere>

[SuiteTarget]
HideDialog=TRUE
TargetDirectory=C:\Program Files\MicroStrategy

[ComponentSelection]
HideDialog=TRUE
### Visible Components ###
ServerAdminVisible=TRUE
CommandManagerVisible=TRUE
EnterpriseManagerVisible=TRUE
ObjectManagerVisible=TRUE
IServerVisible=TRUE
IServerOLAPServicesVisible=TRUE
IServerReportServicesVisible=TRUE
IServerDistributionServicesVisible=TRUE
IServerTransactionServicesVisible=TRUE
IntegrityManagerVisible=TRUE
SystemManagerVisible=TRUE
DeveloperVisible=FALSE
AnalystVisible=FALSE
ArchitectVisible=FALSE
FunctionPluginVisible=FALSE
WebAnalystVisible=FALSE
WebProfessionalVisible=FALSE
WebReporterVisible=FALSE
WebServerASPNETVisible=FALSE
WebServerJSPVisible=FALSE
WebServicesASPNETVisible=FALSE
WebServicesJSPVisible=FALSE
OfficeVisible=FALSE
MobileVisible=FALSE
MobileClientVisible=FALSE
MobileServerASPVisible=FALSE
MobileServerJSPVisible=FALSE
AnalyticsModulesVisible=FALSE
NCSAdminVisible=FALSE
DeliveryEngineVisible=FALSE
SubscriptionPortalVisible=FALSE
TutorialDeliveryInstallVisible=FALSE
TutorialDeliveryConfigureVisible=FALSE
SequeLinkVisible=FALSE
PortletsVisible=FALSE
MDXCubeProviderVisible=FALSE
GISConnectorsVisible=FALSE
OperationsManagerASPVisible=FALSE
OperationsManagerJSPVisible=FALSE
### Components To Install (TRUE) or Remove (FALSE) ###
ServerAdminSelect=TRUE
CommandManagerSelect=TRUE
EnterpriseManagerSelect=TRUE
ObjectManagerSelect=TRUE
IServerSelect=TRUE
IServerOLAPServicesSelect=TRUE
IServerReportServicesSelect=TRUE
IServerDistributionServicesSelect=TRUE
IServerTransactionServicesSelect=TRUE
IntegrityManagerSelect=TRUE
SystemManagerSelect=TRUE
DeveloperSelect=FALSE
AnalystSelect=FALSE
ArchitectSelect=FALSE
FunctionPluginSelect=FALSE
WebAnalystSelect=FALSE
WebProfessionalSelect=FALSE
WebReporterSelect=FALSE
WebServerASPNETSelect=FALSE
WebServerJSPSelect=FALSE
WebServicesASPNETSelect=FALSE
WebServicesJSPSelect=FALSE
OfficeSelect=FALSE
MobileSelect=FALSE
MobileClientSelect=FALSE
MobileServerASPSel ect=FALSE
MobileServerJSPSelect=FALSE
AnalyticsModulesSelect=FALSE
NCAdminSelect=FALSE
DeliveryEngineSelect=FALSE
SubscriptionPortalSelect=FALSE
TutorialDeliveryInstallSelect=FALSE
TutorialDeliveryConfigureSelect=FALSE
SequeLinkSelect=FALSE
PortletsSelect=FALSE
MDXCubeProviderSelect=FALSE
GISConnectorsSelect=FALSE
OperationsManagerASPSel ect=FALSE
OperationsManagerJSPSelect=FALSE
[IServerServiceAccount]
HideDialog=TRUE
SkipAccountSetting=FALSE
Login=<UserName>
Password=<Password>
Domain=<Domain>
ServiceStartUp=AUTO
[Summary]
HideDialog=TRUE
[Finish]
HideDialog=TRUE

Copy and paste this example to create a response.ini file. Replace any text between angled brackets (<>)) with your own specific information. For example, change UserName=<UserNameHere> to UserName=jsmith. Make sure to check that all file paths are entered with correct spacing.

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 IServerOLAPServicesSelect=TRUE and IServerOLAPServicesVisible=TRUE to install these components.

Example of a response.ini file: Installs Intelligence Server components

[Installer]
HideAllDialogs=TRUE
ForceReboot=TRUE
StopAllServices=TRUE
[Welcome]
HideDialog=TRUE
RemoveAll=FALSE
[UserRegistration]
HideDialog=TRUE
UserName=<UserNameHere>
CompanyName=<CompanyNameHere>
LicenseKey=<CustomerLicenseKeyHere>
[SuiteTarget]
HideDialog=TRUE
TargetDirectory=C:\Program Files\MicroStrategy
[ComponentSelection]
HideDialog=TRUE
### Visible Components ###
ServerAdminVisible=TRUE
IServerVisible=TRUE
IServerOLAPServicesVisible=TRUE
IServerReportServicesVisible=TRUE
IServerDistributionServicesVisible=TRUE
IServerTransactionServicesVisible=TRUE
CommandManagerVisible=FALSE
EnterpriseManagerVisible=FALSE
ObjectManagerVisible=FALSE
IntegrityManagerVisible=FALSE
DeveloperVisible=FALSE
AnalystVisible=FALSE
ArchitectVisible=FALSE
FunctionPluginVisible=FALSE
WebAnalystVisible=FALSE
WebProfessionalVisible=FALSE
WebReporterVisible=FALSE
WebServerASPNETVisible=FALSE
WebServerJSPVisible=FALSE
WebServicesASPNETVisible=FALSE
WebServicesJSPVisible=FALSE
OfficeVisible=FALSE
MobileVisible=FALSE
MobileClientVisible=FALSE
MobileServerASPVisible=FALSE
MobileServerJSPVisible=FALSE
AnalyticsModulesVisible=FALSE
NCSAdminVisible=FALSE
DeliveryEngineVisible=FALSE
SubscriptionPortalVisible=FALSE
TutorialDeliveryInstallVisible=FALSE
TutorialDeliveryConfigureVisible=FALSE
SequeLinkVisible=FALSE
PortletsVisible=FALSE
MDXCubeProviderVisible=FALSE
GISConnectorsVisible=FALSE
SystemManagerVisible=FALSE
OperationsManagerASPVisible=FALSE
OperationsManagerJSPVisible=FALSE
### Components To Install (TRUE) or Remove (FALSE) ###
ServerAdminSelect=TRUE
IServerSelect=TRUE
IServerOLAPServicesSelect=TRUE
IServerReportServicesSelect=TRUE
IServerDistributionServicesSelect=TRUE
IServerTransactionServicesSelect=TRUE
CommandManagerSelect=FALSE
EnterpriseManagerSelect=FALSE
ObjectManagerSelect=FALSE
IntegrityManagerSelect=FALSE
DeveloperSelect=FALSE
AnalystSelect=FALSE
ArchitectSelect=FALSE
FunctionPluginSelect=FALSE
WebAnalystSelect=FALSE
WebProfessionalSelect=FALSE
WebReporterSelect=FALSE
WebServerASPNETSelect=FALSE
WebServerJSPSelect=FALSE
WebServicesASPNETSelect=FALSE
WebServicesJSPSelect=FALSE
OfficeSelect=FALSE
MobileSelect=FALSE
MobileClientSelect=FALSE
MobileServerASPSelect=FALSE
MobileServerJSPSelect=FALSE
AnalyticsModulesSelect=FALSE
NCSAdminSelect=FALSE
DeliveryEngineSelect=FALSE
SubscriptionPortalSelect=FALSE
TutorialDeliveryInstallSelect=FALSE
TutorialDeliveryConfigureSelect=FALSE
SequeLinkSelect=FALSE
PortletsSelect=FALSE
MDXCubeProviderSelect=FALSE
GISConnectorsSelect=FALSE
SystemManagerSelect=FALSE
OperationsManagerASPSelect=FALSE
OperationsManagerJSPSelect=FALSE
 [IServerServiceAccount]
HideDialog=TRUE
SkipAccountSetting=FALSE
Login=<UserName>
Password=<Password>
Domain=<Domain>
ServiceStartUp=AUTO
[Summary]
HideDialog=TRUE
[Finish]
HideDialog=TRUE

Copy and paste this example to create a response.ini file. Replace any text between angled brackets (<>) with your own specific information. For example, change UserName=<UserNameHere> to UserName=jsmith. Make sure to check that all file paths are entered with correct spacing.

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 IServerOLAPServicesSelect=TRUE and IServerOLAPServicesVisible=TRUE to install these components.

**Example of a response.ini file: Installs MicroStrategy Web components**

[Installer]
HideAllDialogs=TRUE
ForceReboot=TRUE
StopAllServices=TRUE
StopIIS=TRUE
EnableASPServices=TRUE
EnableASPNETServices=TRUE
[Welcome]
HideDialog=TRUE
RemoveAll=FALSE
[UserRegistration]
HideDialog=TRUE
UserName=UserNameHere
CompanyName=CompanyNameHere
LicenseKey=CustomerLicenseKeyHere
[SuiteTarget]
HideDialog=TRUE
TargetDirectory=C:\Program Files\MicroStrategy
[ComponentSelection]
HideDialog=TRUE
### Visible Components ###
WebReporterVisible=TRUE
WebAnalystVisible=TRUE
WebProfessionalVisible=TRUE
WebServerASPNETVisible=TRUE
WebServerJSPVisible=TRUE
WebServicesASPNETVisible=TRUE
WebServicesJSPVisible=TRUE
ServerAdminVisible=FALSE
IServerVisible=FALSE
IServerOLAPServicesVisible=FALSE
IServerReportServicesVisible=FALSE
IServerDistributionServicesVisible=FALSE
IServerTransactionServicesVisible=FALSE
CommandManagerVisible=FALSE
EnterpriseManagerVisible=FALSE
ObjectManagerVisible=FALSE
IntegrityManagerVisible=FALSE
DeveloperVisible=FALSE
AnalystVisible=FALSE
ArchitectVisible=FALSE
FunctionPluginVisible=FALSE
OfficeVisible=FALSE
MobileVisible=FALSE
MobileClientVisible=FALSE
MobileServerASPVisible=FALSE
MobileServerJSPVisible=FALSE
AnalyticsModulesVisible=FALSE
NCSAdminVisible=FALSE
DeliveryEngineVisible=FALSE
SubscriptionPortalVisible=FALSE
TutorialDeliveryInstallVisible=FALSE
TutorialDeliveryConfigureVisible=FALSE
SequeLinkVisible=FALSE
PortletsVisible=FALSE
MDXCubeProviderVisible=FALSE
GISConnectorsVisible=FALSE
SystemManagerVisible=FALSE
### Components To Install (TRUE) or Remove (FALSE) ###

- WebReporterSelect=TRUE
- WebAnalystSelect=TRUE
- WebProfessionalSelect=TRUE
- WebServerASPNETSelect=TRUE
- WebServerJSPSelect=TRUE
- WebServicesASPNETSelect=TRUE
- WebServicesJSPSelect=TRUE
- ServerAdminSelect=FALSE
- IServerSelect=FALSE
- IServerOLAPServicesSelect=FALSE
- IServerReportServicesSelect=FALSE
- IServerDistributionServicesSelect=FALSE
- IServerTransactionServicesSelect=FALSE
- CommandManagerSelect=FALSE
- EnterpriseManagerSelect=FALSE
- ObjectManagerSelect=FALSE
- IntegrityManagerSelect=FALSE
- DeveloperSelect=FALSE
- AnalystSelect=FALSE
- ArchitectSelect=FALSE
- FunctionPluginSelect=FALSE
- OfficeSelect=FALSE
- MobileSelect=FALSE
- MobileClientSelect=FALSE
- MobileServerASPSelect=FALSE
- MobileServerJSPSelect=FALSE
- AnalyticsModulesSelect=FALSE
- NCSAdminSelect=FALSE
- DeliveryEngineSelect=FALSE
- SubscriptionPortalSelect=FALSE
- TutorialDeliveryInstallSelect=FALSE
- TutorialDeliveryConfigureSelect=FALSE
- SequeLinkSelect=FALSE
- PortletsSelect=FALSE
- MDXCubeProviderSelect=FALSE
- GISConnectorsSelect=FALSE
- SystemManagerSelect=FALSE
- OperationsManagerASPSelect=FALSE
- OperationsManagerJSPSelect=FALSE

[WebVirtualDirectory]
- HideDialog=TRUE
- VirtualDirectory=MicroStrategy
- ReconfigureVirtualDirectory=TRUE

[WebServicesVirtualDirectory]
- HideDialog=TRUE
- VirtualDirectory=MicroStrategyWS
ReconfigureVirtualDirectory=TRUE
[Summary]
HideDialog=TRUE
[Finish]
HideDialog=TRUE

Copy and paste this example to create a response.ini file. Replace any text between angled brackets (<>) with your own specific information. For example, change UserName=<UserNameHere> to UserName=jsmith. Make sure to check that all file paths are entered with correct spacing.

Your license key determines which MicroStrategy components will be available for your installation. For example, if your license key does not include MicroStrategy Web Reporter, then you cannot use WebReporterVisible=TRUE and WebReporterSelect=TRUE to install these components.

Example of a response.ini file: Installs Enterprise Manager components

[Installer]
HideAllDialogs=TRUE
ForceReboot=TRUE
StopAllServices=TRUE
EnterpriseManagerOverwrite=TRUE
[Welcome]
HideDialog=TRUE
RemoveAll=FALSE
[UserRegistration]
HideDialog=TRUE
UserName=<UserNameHere>
CompanyName=<CompanyNameHere>
LicenseKey=<CustomerLicenseKeyHere>
[SuiteTarget]
HideDialog=TRUE
TargetDirectory=C:\Program Files\MicroStrategy
[ComponentSelection]
HideDialog=TRUE
### Visible Components ###
EnterpriseManagerVisible=TRUE
ServerAdminVisible=FALSE
CommandManagerVisible=FALSE
ObjectManagerVisible=FALSE
IServerVisible=FALSE
IServerOLAPServicesVisible=FALSE
IServerReportServicesVisible=FALSE
IServerDistributionServicesVisible=FALSE
IServerTransactionServicesVisible=FALSE
IntegrityManagerVisible=FALSE
DeveloperVisible=FALSE
AnalystVisible=FALSE
ArchitectVisible=FALSE
FunctionPluginVisible=FALSE
WebAnalystVisible=FALSE
WebProfessionalVisible=FALSE
WebReporterVisible=FALSE
WebServerASPNETVisible=FALSE
WebServerJSPVisible=FALSE
WebServicesASPNETVisible=FALSE
WebServicesJSPVisible=FALSE
OfficeVisible=FALSE
MobileVisible=FALSE
MobileClientVisible=FALSE
MobileServerASPVisible=FALSE
MobileServerJSPVisible=FALSE
AnalyticsModulesVisible=FALSE
NCSAdminVisible=FALSE
DeliveryEngineVisible=FALSE
SubscriptionPortalVisible=FALSE
TutorialDeliveryInstallVisible=FALSE
TutorialDeliveryConfigureVisible=FALSE
SequeLinkVisible=FALSE
PortletsVisible=FALSE
MDXCubeProviderVisible=FALSE
GISConnectorsVisible=FALSE
SystemManagerVisible=FALSE
OperationsManagerASPVisible=FALSE
OperationsManagerJSPVisible=FALSE
### Components To Install (TRUE) or Remove (FALSE) ###
EnterpriseManagerSelect=TRUE
ServerAdminSelect=FALSE
CommandManagerSelect=FALSE
ObjectManagerSelect=FALSE
IServerSelect=FALSE
IServerOLAPServicesSelect=FALSE
IServerReportServicesSelect=FALSE
IServerDistributionServicesSelect=FALSE
IServerTransactionServicesSelect=FALSE
IntegrityManagerSelect=FALSE
DeveloperSelect=FALSE
AnalystSelect=FALSE
ArchitectSelect=FALSE
FunctionPluginSelect=FALSE
WebAnalystSelect=FALSE
WebProfessionalSelect=FALSE
WebReporterSelect=FALSE
WebServerASPNETSelect=FALSE
WebServerJSPSelect=FALSE
WebServicesASPNETSelect=FALSE
WebServicesJSPSelect=FALSE
OfficeSelect=FALSE
MobileSelect=FALSE
MobileClientSelect=FALSE
MobileServerASPSelect=FALSE
MobileServerJSPSelect=FALSE
AnalyticsModulesSelect=FALSE
NCSAdminSelect=FALSE
DeliveryEngineSelect=FALSE
SubscriptionPortalSelect=FALSE
TutorialDeliveryInstallSelect=FALSE
TutorialDeliveryConfigureSelect=FALSE
SequeLinkSelect=FALSE
PortletsSelect=FALSE
MDXCubeProviderSelect=FALSE
GISConnectorsSelect=FALSE
SystemManagerSelect=FALSE
OperationsManagerASPSelect=FALSE
OperationsManagerJSPSelect=FALSE
[Summary]
HideDialog=TRUE
[Finish]
HideDialog=TRUE

Copy and paste this example to create a response.ini file. Replace any text between angled brackets (<>)) with your own specific information. For example, change UserName=<UserNameHere> to UserName=jsmith. Make sure to check that all file paths are entered with correct spacing.

Your license key determines which MicroStrategy components will be available for your installation. For example, if your license key does not include MicroStrategy Enterprise Manager, then you cannot use EnterpriseManagerVisible=TRUE and EnterpriseManagerSelect=TRUE to install these components.

Example of a response.ini file: Installs Developer, Architect, and other components

[Installer]
HideAllDialogs=TRUE
ForceReboot=TRUE
StopAllServices=TRUE
EnterpriseManagerOverwrite=TRUE

[Welcome]
HideDialog=TRUE
RemoveAll=FALSE

[UserRegistration]
HideDialog=TRUE
UserName=<UserNameHere>
CompanyName=<CompanyNameHere>
LicenseKey=<CustomerLicenseKeyHere>

[SuiteTarget]
HideDialog=TRUE
TargetDirectory=C:\Program Files\MicroStrategy [ComponentSelection]
HideDialog=TRUE
### Visible Components ###
AnalystVisible=TRUE
DeveloperVisible=TRUE
ArchitectVisible=TRUE
CommandManagerVisible=TRUE
EnterpriseManagerVisible=TRUE
ObjectManagerVisible=TRUE
FunctionPluginVisible=TRUE
ServerAdminVisible=FALSE
IServerVisible=FALSE
IServerOLAPServicesVisible=FALSE
IServerReportServicesVisible=FALSE
IServerDistributionServicesVisible=FALSE
IServerTransactionServicesVisible=FALSE
IntegrityManagerVisible=FALSE
Web AnalystVisible=FALSE
Web ProfessionalVisible=FALSE
Web ReporterVisible=FALSE
WebServerASPNETVisible=FALSE
WebServerJSPVisible=FALSE
WebServicesASPNETVisible=FALSE
WebServicesJSPVisible=FALSE
OfficeVisible=FALSE
MobileVisible=FALSE
Mobile ClientVisible=FALSE
Mobile ServerASPVisible=FALSE
Mobile ServerJSPVisible=FALSE
AnalyticsModulesVisible=FALSE
NCSAdminVisible=FALSE
DeliveryEngineVisible=FALSE
SubscriptionPortalVisible=FALSE
TutorialDeliveryInstallVisible=FALSE
TutorialDeliveryConfigureVisible=FALSE
SequeLinkVisible=FALSE
PortletsVisible=FALSE
MDXCubeProviderVisible=FALSE
GISConnectorsVisible=FALSE
SystemManagerVisible=FALSE
OperationsManagerASPVisible=FALSE
OperationsManagerJSPVisible=FALSE
### Components To Install (TRUE) or Remove (FALSE) ###
AnalystSelect=TRUE
DeveloperSelect=TRUE
ArchitectSelect=TRUE
CommandManagerSelect=TRUE
EnterpriseManagerSelect=TRUE
ObjectManagerSelect=TRUE
FunctionPluginSelect=TRUE
ServerAdminSelect=FALSE
IServerSelect=FALSE
IServerOLAPServicesSelect=FALSE
IServerReportServicesSelect=FALSE
IServerDistributionServicesSelect=FALSE
IServerTransactionServicesSelect=FALSE
IntegrityManagerSelect=FALSE
WebAnalystSelect=FALSE
WebProfessionalSelect=FALSE
WebReporterSelect=FALSE
WebServerASPNETSelect=FALSE
WebServerJSPSelect=FALSE
WebServicesASPNETSelect=FALSE
WebServicesJSPSelect=FALSE
OfficeSelect=FALSE
MobileSelect=FALSE
MobileClientSelect=FALSE
MobileServerASPSelect=FALSE
MobileServerJSPSelect=FALSE
AnalyticsModulesSelect=FALSE
NCSAdminSelect=FALSE
DeliveryEngineSelect=FALSE
SubscriptionPortalSelect=FALSE
TutorialDeliveryInstallSelect=FALSE
TutorialDeliveryConfigureSelect=FALSE
SequeLinkSelect=FALSE
PortletsSelect=FALSE
MDXCubeProviderSelect=FALSE
GISConnectorsSelect=FALSE
SystemManagerSelect=FALSE
OperationsManagerASPISelect=FALSE
OperationsManagerJSPSelect=FALSE

[Summary]
HideDialog=TRUE
[Finish]
HideDialog=TRUE

Copy and paste this example to create a response.ini file. Replace any text between angled brackets (<>) with your own specific information. Make sure to check that all file paths are entered with correct spacing.

Your license key determines which MicroStrategy components will be available for your installation. For example, if your license key does not include MicroStrategy Developer, then you cannot use DeveloperVisible=TRUE and DeveloperSelect=TRUE to install these components.
Example of a response.ini file: Installs Developer components

[Installer]
HideAllDialogs=TRUE
ForceReboot=TRUE
StopAllServices=TRUE
[Welcome]
HideDialog=TRUE
RemoveAll=FALSE
[UserRegistration]
HideDialog=TRUE
UserName=<UserNameHere>
CompanyName=<CompanyNameHere>
LicenseKey=<CustomerLicenseKeyHere>
[SuiteTarget]
HideDialog=TRUE
TargetDirectory=C:\Program Files\MicroStrategy
[ComponentSelection]
HideDialog=TRUE
### Visible Components ###
AnalystVisible=TRUE
DeveloperVisible=TRUE
ArchitectVisible=FALSE
FunctionPluginVisible=FALSE
CommandManagerVisible=FALSE
EnterpriseManagerVisible=FALSE
ObjectManagerVisible=FALSE
ServerAdminVisible=FALSE
IServerVisible=FALSE
IServerOLAPServicesVisible=FALSE
IServerReportServicesVisible=FALSE
IServerDistributionServicesVisible=FALSE
IServerTransactionServicesVisible=FALSE
IntegrityManagerVisible=FALSE
WebAnalystVisible=FALSE
WebProfessionalVisible=FALSE
WebReporterVisible=FALSE
WebServerASPNETVisible=FALSE
WebServerJSPVisible=FALSE
WebServicesASPNETVisible=FALSE
WebServicesJSPVisible=FALSE
OfficeVisible=FALSE
MobileVisible=FALSE
MobileClientVisible=FALSE
MobileServerASPVisible=FALSE
MobileServerJSPVisible=FALSE
AnalyticsModulesVisible=FALSE
NCAdminVisible=FALSE
SubscriptionPortalVisible=FALSE
Components To Install (TRUE) or Remove (FALSE)

- AnalystSelect=TRUE
- DeveloperSelect=TRUE
- ArchitectSelect=FALSE
- FunctionPluginSelect=FALSE
- CommandManagerSelect=FALSE
- EnterpriseManagerSelect=FALSE
- ObjectManagerSelect=FALSE
- ServerAdminSelect=FALSE
- IServerSelect=FALSE
- IServerOLAPServicesSelect=FALSE
- IServerReportServicesSelect=FALSE
- IServerDistributionServicesSelect=FALSE
- IServerTransactionServicesSelect=FALSE
- IntegrityManagerSelect=FALSE
- WebAnalystSelect=FALSE
- WebProfessionalSelect=FALSE
- WebReporterSelect=FALSE
- WebServerASPNETSelect=FALSE
- WebServerJSPSelect=FALSE
- WebServicesASPNETSelect=FALSE
- WebServicesJSPSelect=FALSE
- OfficeSelect=FALSE
- MobileSelect=FALSE
- MobileClientSelect=FALSE
- MobileServerASPSelect=FALSE
- MobileServerJSPSelect=FALSE
- AnalyticsModulesSelect=FALSE
- NCSAdminSelect=FALSE
- DeliveryEngineSelect=FALSE
- SubscriptionPortalSelect=FALSE
- TutorialDeliveryInstallSelect=FALSE
- TutorialDeliveryConfigureSelect=FALSE
- SequeLinkSelect=FALSE
- PortletsSelect=FALSE
- MDXCubeProviderSelect=FALSE
- GISConnectorsSelect=FALSE
- SystemManagerSelect=FALSE
- OperationsManagerASPSelect=FALSE
- OperationsManagerJSPSelect=FALSE
[Summary]
HideDialog=TRUE
[Finish]
HideDialog=TRUE

Copy and paste this example to create a response.ini file. Replace any text between angled brackets (<>) with your own specific information. Make sure to check that all file paths are entered with correct spacing.

Your license key determines which MicroStrategy components will be available for your installation. For example, if your license key does not include MicroStrategy Developer, then you cannot use DeveloperVisible=TRUE and DeveloperSelect=TRUE to install these components.

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.

   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.

   - Press ENTER to begin the installation.

### Activating your installation

After your installation is complete, you must activate your MicroStrategy software installation within 30 days. To activate your software you can follow the instructions provided in *Chapter 4, Activating Your Installation*.

### 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 164*.

### 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:

```ini
[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 268.*

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 347.*

**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.
### Function
Identifies the function calls in the setup script.

### Information
Logs information about the setup that is running.

### Warning
Includes feedback that you must verify related to the setup. For 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
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][1: 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 silent installations
After your silent installation is complete, you must activate your MicroStrategy software installation within 30 days. To activate your software you can follow the instructions provided in *Chapter 4, 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 if you are prompted to stop your Intelligence Server or your IIS Web server. If this happens, check:

- The version of Intelligence Server to ensure that you have the right one for the products you are installing
- That 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 347. This way you can change the settings in the response.ini file without having to generate a new .iss file.

**Silent installation of a hotfix**

A hotfix is a file or a collection of files that you can apply to MicroStrategy products installed on your computer to correct a specific problem. Hotfixes are packaged in an executable (.exe) file, which is a self-installing format. When you install a hotfix, files are backed up automatically so that you can remove the hotfix later.

You can apply a hotfix to MicroStrategy products using silent installation. For information on installing the hotfix of MicroStrategy Office, see Silent installation of MicroStrategy Office, page 318.

**Prerequisites**

- You must obtain and unzip the MicroStrategy hotfix files. These files include HFResponse.ini and HFsetup.iss. The rest of this procedure assumes you have saved these files to the file path C:\. If you save it to another file path, replace C:\ with the file path to the HFResponse.ini and HFsetup.iss.

**To perform a silent installation of a hotfix**

If you are installing a hotfix of MicroStrategy Office, see Silent installation of MicroStrategy Office, page 318.

1. Insert the disk containing the Hotfix installable in the disk drive.
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.
3. 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. Access the disk drive through the command prompt.
6. Run the silent installation with the following command:

   InstallPath\setup.exe -LLanguageValue --ResponseFile="C:\HFresponse.ini" -s -f1"C:\HFsetup.iss"
In the command listed above, you must supply the following information:

- **InstallPath**: The location in which to install the MicroStrategy hotfix.

- **LanguageValue**: Determines the language for the installation. Refer to *Language settings for silent installations, page 350*, for an explanation of the LanguageValue parameter. For example, to select English, the command is as follows: `InstallPath\setup.exe -LL0009 --ResponseFile="C:\HFresponse.ini" -s -f"C:\HFsetup.iss"

You can check the status of the silent installation in Windows Task Manager. After applying the Hotfix, you can view it in the Add/Remove Programs window.

**Silent installation of MicroStrategy Office**

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 hotfixes 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 Office User Guide*.

**Prerequisites**

- 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, 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.

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 hotfix 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 48* in *Chapter 1, Planning Your Installation*, for installation and configuration checklists.
AUTOMATED INSTALLATION ON UNIX AND LINUX

This chapter explains the various methods of performing a fully automated and unattended installation within the MicroStrategy platform when you do not have access to a UNIX or Linux graphical user interface (GUI).

Intelligence Server configurations possible through the command line on UNIX and Linux are covered in Chapter 11, Configuring MicroStrategy Using Command Line Tools.

This chapter has one section: Silent installation, page 321

Before installing MicroStrategy products, you should refer to 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 322 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 325.
To complete a silent installation

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.

3. Browse to the MicroStrategyInstallation/QueryReportingAnalysis_UNIX directory. Depending on your UNIX or Linux environment, browse to one of the following folders:
   - **Solaris**: QueryReportingAnalysis_SunOS
   - **AIX**: QueryReportingAnalysis_AIX
   - **HP-UX**: QueryReportingAnalysis_HP-UX
   - **Linux**: QueryReportingAnalysis_Linux

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 325.

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:

   ```bash
   setup.sh -silent -options options.txt
   ```

   If you are installing a MicroStrategy hotfix, use the setupHF.sh file instead. For example, type the command:

   ```bash
   setupHF.sh -silent -options options.txt
   ```

**Example of an options file**

Copy and paste this example to create an options.txt file. Replace any italicized text with your own specific information. For example, change `userRegistration.user=UserName` to `userRegistration.user=jsmith`. 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 325.

⚠️ Be aware of the following:
• The example below assumes you are not using a CPU-based license for MicroStrategy Intelligence Server. For CPU-based licenses, you must add the parameter:

    cpuCount.number=maximum

• 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 Providing installation files for smaller installations, page 339.

    licenseAgreement.accept=true
    userRegistration.user=UserName
    userRegistration.company=CompanyName
    userRegistration.cdKey=License
    install.Instance=new
    install.Operation=FRESH_INSTALL
    suite.homeDirectory=Path
    suite.installDirectory=Path
    suite.logDirectory=Path
    SelectComponents.visible=true
    IntelligenceServer.visible=true
    IntelligenceServer.active=true
    ReportServices.visible=true
    ReportServices.active=true
    OLAPServices.visible=true
    OLAPServices.active=true
    DistributionServices.visible=true
    DistributionServices.active=true
    TransactionServices.visible=true
    TransactionServices.active=true
    WebAnalyst.visible=true
    WebAnalyst.active=true
    WebReporter.visible=true
    WebReporter.active=true
    WebProfessional.visible=true
    WebProfessional.active=true
    Portlets.visible=true
    Portlets.active=true
    GISConnectors.visible=true
    GISConnectors.active=true
    WebServices.visible=true
    WebServices.active=true
    MobileClient.visible=true
    MobileClient.active=true
    MobileServer.visible=true
    MobileServer.active=true
    CommandManager.visible=true
    CommandManager.active=true
    EnterpriseManager.visible=true
EnterpriseManager.active=true
SystemManager.visible=true
SystemManager.active=true
IntegrityManager.visible=true
IntegrityManager.active=true
OperationsManager.visible=true
OperationsManager.active=true
UsherSecurityServer.visible=true
UsherSecurityServer.active=true
UsherNetworkManager.visible=true
UsherNetworkManager.active=true
UsherAnalytics.visible=true
UsherAnalytics.active=true
UsherMobile.visible=true
UsherMobile.active=true
WebUniversal.deployDirectory=Path
Portlets.installDirectory=Path
GISConnectors.installDirectory=Path
OperationsManager.installDirectory=Path
WebServices.installDirectory=Path
MobileServer.installDirectory=Path
CommandManager.installDirectory=Path
SystemManager.installDirectory=Path
UsherSecurityServer.installDirectory=Path
UsherNetworkManager.installDirectory=Path
UsherAnalytics.installDirectory=Path
UsherMobile.installDirectory=Path
HealthCenter.port=PortNumber
HealthCenter.accessCode=AccessCode
HealthCenter.configureDaemon=true
HealthCenter.master=false
HealthCenter.repository=Path
HealthCenter.customerExperienceProgram=false
UsherSecurityServer.tomcatDir=Path
UsherSecurityServer.serverDBHost=IP_Address
UsherSecurityServer.serverDBPort=Port
UsherSecurityServer.serverDBUser=UserName
UsherSecurityServer.serverDBPassword=Password
UsherSecurityServer.serverDBInstance=DatabaseInstance
UsherSecurityServer.serverLogDBInstance=DatabaseInstance
UsherSecurityServer.serverPortOne=Port
UsherSecurityServer.serverPortTwo=Port
UsherSecurityServer.serverSslCert=Path
UsherSecurityServer.serverSslKey=Path
UsherSecurityServer.serverCaCert=Path
UsherSecurityServer.gatewayPort=Port
UsherSecurityServer.gatewaySslCert=Path
UsherSecurityServer.gatewaySslKey=Path
UsherSecurityServer.gatewayCaCert=Path
UsherNetworkManager.apacheDir=Path
UsherNetworkManager.apacheUser=UserName
UsherNetworkManager.apacheGroup=GroupName
UsherNetworkManager.useSameDBSetting=false
UsherNetworkManager.DBHost=IP_Address
UsherNetworkManager.DBPort=Port
UsherNetworkManager.DBUser=UserName
UsherNetworkManager.DBPassword=UserPassword
UsherNetworkManager.DBInstance=DatabaseInstance
UsherAnalytics.useSameDBSetting=false
UsherAnalytics.DBHost=IP_Address
UsherAnalytics.DBPort=Port
UsherAnalytics.DBUser=UserName
UsherAnalytics.DBPassword=UserPassword

**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:
  ```
  install.Operation=REPAIR
  ```

- To uninstall an existing MicroStrategy installation:
  ```
  install.Operation=UNINSTALL
  ```

- To upgrade an existing MicroStrategy installation:
  ```
  install.Operation=UPGRADE
  ```

- To install a MicroStrategy hotfix installation:
  ```
  install.Operation=HOTFIX_INSTALL
  ```

- To uninstall an existing MicroStrategy hotfix installation:
  ```
  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 *Choose Destination Location, page 86.*

- Home directory: The location where the MicroStrategy configuration files and application launchers are to be installed.
suite.homeDirectory=Path

- Install directory: The location where the MicroStrategy products are to be installed.
  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:

- Each product includes 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 you do not specify a *true* or *false* value for each product, *true* is used 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.

- Each product includes 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 1*.

**MicroStrategy Intelligence Server**

The state of whether MicroStrategy Intelligence Server is displayed in the MicroStrategy Installation Wizard:

IntelligenceServer.visible=Value

The selection state of MicroStrategy Intelligence Server.

IntelligenceServer.active=Value

For example, to select MicroStrategy Intelligence Server for installation, use:

IntelligenceServer.visible=true
IntelligenceServer.active=true

MicroStrategy Intelligence Server is installed if you select to install any of the following MicroStrategy products:

- *MicroStrategy Report Services, page 328*
- *MicroStrategy OLAP Services, page 328*
- *MicroStrategy Distribution Services, page 328*
- *MicroStrategy Transaction Services, page 329*

**MicroStrategy Report Services**

The state of whether MicroStrategy Report Services is displayed in the MicroStrategy Installation Wizard:

```
ReportServices.visible=Value
```

The selection state of MicroStrategy Report Services.

```
ReportServices.active=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=Value

The selection state of MicroStrategy Transaction Services.
  TransactionServices.active=Value

For example, to select MicroStrategy Transaction Services for installation, use:
  TransactionServices.visible=true
  TransactionServices.active=true

**MicroStrategy Web Analyst**

The state of whether MicroStrategy Web Analyst is displayed in the MicroStrategy Installation Wizard:
  WebAnalyst.visible=Value

The selection state of MicroStrategy Web Analyst.
  WebAnalyst.active=Value

For example, to select MicroStrategy Web Analyst for installation, use:
  WebAnalyst.visible=true
  WebAnalyst.active=true

**MicroStrategy Web Reporter**

The state of whether MicroStrategy Web Reporter is displayed in the MicroStrategy Installation Wizard:
  WebReporter.visible=Value

The selection state of MicroStrategy Web Reporter.
  WebReporter.active=Value

For example, to select MicroStrategy Web Reporter for installation, use:
WebReporter.visible=true
WebReporter.active=true

**MicroStrategy Web Professional**

The state of whether MicroStrategy Web Professional is displayed in the MicroStrategy Installation Wizard:

WebProfessional.visible=Value

The selection state of MicroStrategy Web Professional.

WebProfessional.active=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=Value

The selection state of MicroStrategy Portlets.

Portlets.active=Value

For example, to select MicroStrategy Portlets for installation, use:

Portlets.visible=true
Portlets.active=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:

```plaintext
WebServices.visible=Value
```

The selection state of MicroStrategy Web Services J2EE.

```plaintext
WebServices.active=Value
```

For example, to select MicroStrategy Web Services J2EE for installation, use:

```plaintext
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:

```plaintext
MobileServer.visible=Value
```

The selection state of Mobile Server JSP.

```plaintext
MobileServer.active=Value
```

For example, to select MicroStrategy Mobile Server JSP for installation, use:

```plaintext
MobileServer.visible=true
MobileServer.active=true
```

**MicroStrategy Command Manager**

The state of whether MicroStrategy Command Manager is displayed in the MicroStrategy Installation Wizard:

```plaintext
CommandManager.visible=Value
```

The selection state of MicroStrategy Command Manager.

```plaintext
CommandManager.active=Value
```

For example, to select MicroStrategy Command Manager for installation, use:

```plaintext
CommandManager.visible=true
CommandManager.active=true
```
**MicroStrategy System Manager**

The state of whether MicroStrategy System Manager is displayed in the MicroStrategy Installation Wizard:

\[ \text{SystemManager.visible} = \text{Value} \]

The selection state of MicroStrategy System Manager.

\[ \text{SystemManager.active} = \text{Value} \]

For example, to select MicroStrategy System Manager for installation, use:

\[ \text{SystemManager.visible} = \text{true} \]
\[ \text{SystemManager.active} = \text{true} \]

**MicroStrategy Integrity Manager**

The state of whether MicroStrategy Integrity Manager is displayed in the MicroStrategy Installation Wizard:

\[ \text{IntegrityManager.visible} = \text{Value} \]

The selection state of MicroStrategy Integrity Manager.

\[ \text{IntegrityManager.active} = \text{Value} \]

For example, to select MicroStrategy Integrity Manager for installation, use:

\[ \text{IntegrityManager.visible} = \text{true} \]
\[ \text{IntegrityManager.active} = \text{true} \]

**MicroStrategy Enterprise Manager**

The state of whether MicroStrategy Enterprise Manager is displayed in the MicroStrategy Installation Wizard:

\[ \text{EnterpriseManager.visible} = \text{Value} \]

The selection state of MicroStrategy Enterprise Manager.

\[ \text{EnterpriseManager.active} = \text{Value} \]

For example, to select MicroStrategy Enterprise Manager for installation, use:

\[ \text{EnterpriseManager.visible} = \text{true} \]
\[ \text{EnterpriseManager.active} = \text{true} \]
**MicroStrategy Operations Manager**

The state of whether MicroStrategy Operations Manager is displayed in the MicroStrategy Installation Wizard:

\[
\text{OperationsManager.visible} = \text{Value}
\]

The selection state of MicroStrategy Operations Manager.

\[
\text{OperationsManager.active} = \text{Value}
\]

For example, to select MicroStrategy Operations Manager for installation, use:

\[
\text{OperationsManager.visible} = \text{true}
\]
\[
\text{OperationsManager.active} = \text{true}
\]

**Usher Security Server**

The state of whether Usher Security Server is displayed in the MicroStrategy Installation Wizard:

\[
\text{UsherSecurityServer.visible} = \text{Value}
\]

The selection state of Usher Security Server.

\[
\text{UsherSecurityServer.active} = \text{Value}
\]

For example, to select Usher Security Server for installation, use:

\[
\text{UsherSecurityServer.visible} = \text{true}
\]
\[
\text{UsherSecurityServer.active} = \text{true}
\]

**Usher Network Manager**

The state of whether Usher Network Manager is displayed in the MicroStrategy Installation Wizard:

\[
\text{UsherNetworkManager.visible} = \text{Value}
\]

The selection state of Usher Network Manager.

\[
\text{UsherNetworkManager.active} = \text{Value}
\]

For example, to select Usher Network Manager for installation, use:

\[
\text{UsherNetworkManager.visible} = \text{true}
\]
\[
\text{UsherNetworkManager.active} = \text{true}
\]
**Usher Analytics**

The state of whether Usher Analytics is displayed in the MicroStrategy Installation Wizard:

```plaintext
UsherAnalytics.visible=Value
```

The selection state of Usher Analytics.

```plaintext
UsherAnalytics.active=Value
```

For example, to select Usher Analytics for installation, use:

```plaintext
UsherAnalytics.visible=true
UsherAnalytics.active=true
```

**Usher Mobile**

The state of whether Usher Mobile is displayed in the MicroStrategy Installation Wizard:

```plaintext
UsherMobile.visible=Value
```

The selection state of Usher Mobile.

```plaintext
UsherMobile.active=Value
```

For example, to select Usher Mobile for installation, use:

```plaintext
UsherMobile.visible=true
UsherMobile.active=true
```

**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.

```plaintext
cpuCount.number=Value
```

By default, the maximum number of CPUs is allowed. This is represented with the following value for this parameter:

```plaintext
cpuCount.number=maximum
```

**MicroStrategy product and component installation locations**

You can define the installation locations for the following products and components:

- MicroStrategy Web Universal Install Location:
  ```plaintext
  WebUniversal.deployDirectory=Value
  ```

- MicroStrategy Portlets Install Location:
Portlets.installDirectory=\Value
- MicroStrategy GIS Connectors Install Location
  GISConnectors.installDirectory=\Value
- MicroStrategy Operations Manager Install Location
  OperationsManager.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
- Usher Security Server Install Location:
  UsherSecurityServer.installDirectory=\Value
- Usher Network Manager Install Location:
  UsherNetworkManager.installDirectory=\Value
- Usher Analytics Install Location:
  UsherAnalytics.installDirectory=\Value
- Usher Mobile Install Location:
  UsherMobile.installDirectory=\Value

**Health Center Agent configuration**

You can configure the machine as a Health Center Agent or Master Health Agent as part of the silent installation:

- The port used by the Health Center Agent:
  HealthCenter.port=\Value
  Provide a valid, available port value between 1,024 and 65,533.
- The access code required to access the Health Center Agent and its information:
  HealthCenter.accessCode=\Value
If no access code is provided, then the Health Center Agent can be accessed without supplying an access code.

- Configure the Health Center Agent as a daemon:
  
  `HealthCenter.configureDaemon=Value`

  You can define this parameter as true or false.

  - **HealthCenter.configureDaemon=true**
    
    Configures this Health Agent as a daemon, so that the Health Agent process is constantly running in the background. This requires you to configure the Health Agent using an account that has root access privileges to the machine.

  - **HealthCenter.configureDaemon=false**
    
    Configures the Health Agent as an application, which is required if you do not have root access to the machine. In this case, be careful not to stop the Health Agent process, so that the machine can remain part of the Health Center system at all times.

- Configure the Master Health Agent:
  
  `HealthCenter.master=Value`

  You can define this parameter as true or false, with the value of true configuring the machine as a Master Health Agent, which is responsible for most of the Health Center operations, such as scheduling system checks and transmitting diagnostics packages to MicroStrategy Technical Support. If you configure the machine as a Master Health Agent, you must specify the following configurations:

  - The location to store the Health Center repository. The repository contains configuration information about the Health Center system, such as the list of machines on the network and the MicroStrategy products they have installed, and also the destination for all exported diagnostics packages:
    
    `HealthCenter.repository=Value`

  - You can choose to enroll the installation in the Customer Experience Improvement Program:
    
    - **HealthCenter.customerExperienceProgram=true**
      
      Enrolls the installation in the Customer Experience Improvement Program. Once enrolled, Health Center transmits anonymous data about your system to MicroStrategy. No report data or prompt answers are collected or transmitted. All information sent to MicroStrategy as a result of this program is stored in the Census subfolder of the Health Center Repository.
    
    - **HealthCenter.customerExperienceProgram=false**
      
     Opts out of the Customer Experience Improvement Program.
**Usher Security Server configuration: Tomcat database**

Usher Security Server installs a database that is a system of record for individual Usher identities. You can configure Usher Security Server to communicate with the database, as part of the silent installation:

- The location of the Tomcat directory used by Usher Security 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:
  
  `UsherSecurityServer.serverDBUser=UserName`

- The password for the database user specified above:
  
  `UsherSecurityServer.serverDBPassword=UserPassword`

- The name of the Usher Security Server database:
  
  `UsherSecurityServer.serverDBInstance=DatabaseInstance`

- The name of the database that stores log information for the Usher Security Server:
  
  `UsherSecurityServer.serverLogDBInstance=DatabaseInstance`

**Usher Security Server configuration: Ports and certificates**

Set up a trust relationship for Usher Security Server using the Public Key Infrastructure (PKI), as part of the silent installation:

- The port used by Usher Security Server for server (one-way SSL) authentication:
  
  `UsherSecurityServer.serverPortOne=Port`

- The port used by Usher Security 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

**Usher Security 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 Usher Security 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

**Usher Network Manager configuration**

You can configure Usher Network Manager as part of the silent installation:

• The location of the Apache directory used by Usher Network 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:
  UsherNetworkManager.apacheUser=UserName

• The Apache group name:
  UsherNetworkManager.apacheGroup=GroupName

• Specify whether to use the same database connection as Usher Security Server:
  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:
UsherNetworkManager.DBHost=IP_Address

- The port number for the database connection:
  UsherNetworkManager.DBPort=Port

- The account name for the database user that administers the database:
  UsherNetworkManager.DBUser=UserName

- The password for the database user specified above:
  UsherNetworkManager.DBPassword=UserPassword

- The name of the Usher Network Manager database:
  UsherNetworkManager.DBInstance=DatabaseInstance

**Usher Analytics configuration**

Usher Analytics installs a database to store Usher activity data. You can configure Usher Analytics to communicate with the database, as part of the silent installation:

- Specify whether to use the same database connection as Usher Security Server:
  UsherAnalytics.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:
  UsherAnalytics.DBHost=IP_Address

- The port number for the database connection:
  UsherAnalytics.DBPort=Port

- The account name for the database user that administers the database:
  UsherAnalytics.DBUser=UserName

- The password for the database user specified above:
  UsherAnalytics.DBPassword=UserPassword

The Usher Analytics database needs to be on the same MySQL instance as the Usher Security Server database.

**Providing installation files for smaller installations**

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 hotfix installation, you must include all of the files provided as part of the hotfix installation in their default location. This means that you cannot use the options provided below to point to the location of the hotfix 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 46. 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=**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 UNIX and 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 97* in Chapter 3, *Installing MicroStrategy on UNIX and 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 266* 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 your silent installation is complete, you must activate your MicroStrategy installation within 30 days. To activate your installation you can follow the instructions provided in *Chapter 4, 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 MicroStrategy on UNIX and Linux machines is covered in the *Configuring MicroStrategy with a response.ini file, page 357* section in Chapter 11, *Configuring MicroStrategy Using Command Line Tools*. 
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 UNIX and Linux, page 50* 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 6, Deploying MicroStrategy Web, Mobile Server, and Operations Manager. For information on customizing MicroStrategy Web, see Customizing MicroStrategy Web, page 346.

- 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 table below provides a best practices checklist of how to deploy MicroStrategy as an OEM application. In addition to using the checklist below to deploy a MicroStrategy as an OEM application, you can also use MicroStrategy System Manager. System Manager lets you 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.

<table>
<thead>
<tr>
<th>Complete</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Install MicroStrategy on an OEM environment. For information on installing MicroStrategy, see:</td>
</tr>
<tr>
<td></td>
<td>• Chapter 2, Installing MicroStrategy on Windows</td>
</tr>
<tr>
<td></td>
<td>• Chapter 3, Installing MicroStrategy on UNIX and Linux</td>
</tr>
<tr>
<td></td>
<td>Create DSNs using the Connectivity Wizard, as described in Creating DSNs for OEM environments, page 344.</td>
</tr>
<tr>
<td></td>
<td>Configure MicroStrategy using the Configuration Wizard. This tool allows you to save configurations as response files that can be used to automate the configuration for the</td>
</tr>
<tr>
<td>Complete</td>
<td>Task</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>OEM deployment. This allows you to re-use all the configurations performed when developing an OEM application for the deployment process as well, as described in Configuring a MicroStrategy installation, page 345.</td>
</tr>
<tr>
<td></td>
<td>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 345.</td>
</tr>
<tr>
<td></td>
<td>Deploy a MicroStrategy OEM application on an OEM's customer environment, as described in Deploying a MicroStrategy OEM application, page 346.</td>
</tr>
<tr>
<td></td>
<td>Create DSNs on the OEM’s customer environment as necessary, as described in Creating DSNs for OEM environments, page 344.</td>
</tr>
<tr>
<td></td>
<td>Configure and tune an OEM deployment through the use of various MicroStrategy tools, as described in Tuning an OEM deployment, page 351.</td>
</tr>
<tr>
<td></td>
<td>If you are modifying a project that has already been deployed as an OEM application, see Updating OEM applications, page 352 for best practices on how to incorporate any custom reports or objects that may have been created for the deployed application.</td>
</tr>
<tr>
<td></td>
<td>Troubleshoot your MicroStrategy OEM applications using MicroStrategy Health Center, as described in Troubleshooting support for MicroStrategy OEM applications, page 353</td>
</tr>
</tbody>
</table>

### 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 131.

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 345.

A DSN can be created using the MicroStrategy Connectivity Wizard, as described in Defining DSNs, page 135.

**Creating DSNs as part of an OEM deployment**

As part of the deployment of an OEM application, the Connectivity Wizard can also be run from the command line to create DSNs on UNIX and Linux environments. This allows you to perform this configuration using scripts. For information on creating DSNs using the command line version of the Connectivity Wizard, see Creating a DSN for a data source, page 354.

For OEM deployments on Windows machines, use the Connectivity Wizard interface to create DSNs, as described in Defining DSNs, page 135.

**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 140.

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 164.

**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 352* 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.

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 [https://resource.microstrategy.com/msdz/default.asp](https://resource.microstrategy.com/msdz/default.asp). You can also access the MicroStrategy Developer Library from the MicroStrategy support site.

### Deploying a MicroStrategy OEM application

Once an OEM application is developed, it then must be deployed to the customer’s environment. Steps to deploy an OEM 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 347*. To use the Installation Wizard to install MicroStrategy products, see:
   • *Chapter 2, Installing MicroStrategy on Windows*
2 Additional configurations are required, as described in Configuring an OEM deployment installation, page 351.

**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 8, 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 UNIX or Linux environment, see Silent installation, page 321.

**Prerequisites**

- 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 267.

<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>
</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:

```plaintext
[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 product#
Company=MicroStrategy
Lang=LanguageValue
[{8CCF3F6C-55B7-4A27-8C68-ADF21D0585A2}-DlgOrder]
Count=0
```

You must save the `setup.iss` file as ANSI encoding.
Be aware of the following

The version in the setup.iss file must match the MicroStrategy version you are installing exactly. For example, if you are installing version 9.3.0 you must enter Version=9.3.0. Entering a version as Version=9 or Version=9.3.x causes an error when trying to perform a silent installation of version 9.3.0.

Refer to *Language settings for silent installations, page 350*, for an explanation of the LanguageValue parameter within the line Lang=LanguageValue.

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:

   Refer to *Language settings for silent installations, page 350*, for an explanation of the LanguageValue parameter.

   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 two 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>
<tr>
<td>French</td>
<td>0012</td>
</tr>
<tr>
<td>German</td>
<td>0007</td>
</tr>
<tr>
<td>Italian</td>
<td>0016</td>
</tr>
<tr>
<td>Japanese</td>
<td>0017</td>
</tr>
<tr>
<td>Korean</td>
<td>0018</td>
</tr>
<tr>
<td>Portuguese</td>
<td>0022</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>2052</td>
</tr>
<tr>
<td>Spanish</td>
<td>0010</td>
</tr>
<tr>
<td>Swedish</td>
<td>0029</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>1028</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 OEM application.
- Create separate DSNs to connect to the data warehouse and the metadata, as described in Creating DSNs for OEM environments, page 344.
- 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 345.
- Configure and tune an OEM deployment through the use of various MicroStrategy tools, as described in Tuning an OEM deployment, page 351.

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-premise 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 Supplemental Reference for System Administration.
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 UNIX and Linux machines is covered in Configuring MicroStrategy with a response.ini file, page 357.

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 OEM application must then be deployed. There are two ways in which an updated OEM application can be redeployed, as described below:

- Replace the entire project in the OEM application, as described in Replacing a project in an OEM application, page 352.
- Merge new objects into the existing project for the OEM application, as described in Merging new objects into a project in an OEM application, page 353.

Replacing a project in an OEM application

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 new 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 Object Manager or 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 Supplemental Reference for System Administration.

**Troubleshooting support for MicroStrategy OEM applications**

MicroStrategy Health Center can help you diagnose and fix problems in your MicroStrategy system. It detects known problems and provides an immediate solution. In cases where Health Center cannot fix a problem immediately, it enables you to bundle relevant log files into a diagnostic package and transmit this package to MicroStrategy Technical Support for review and troubleshooting.

Health Center is provided with a MicroStrategy installation.

For information on using Health Center to diagnose and fix problems in your MicroStrategy environment, refer to the System Administration Guide.
MicroStrategy tools are provided in command line mode on UNIX and 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 UNIX or 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 375.

This chapter covers the configurations listed below:

- **Creating a DSN for a data source, page 354**
- **Testing ODBC connectivity**
- **Configuring MicroStrategy with a response.ini file, page 357**
- **Configuring and controlling Intelligence Server, page 371**

### 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 UNIX and Linux.

You can create DSNs using the MicroStrategy Connectivity Wizard on Windows, UNIX, and Linux machines, as described in Communicating with databases, page 131.

To create a DSN on UNIX/Linux from the command line

1. From a UNIX/Linux console window, browse to HOME_PATH, where HOME_PATH is the directory that you specified as the home directory during installation.

2. Browse to the folder bin.

3. Type `mstrconnectwiz -h`, and then press 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:OracleUserPasword
   ```

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 mstrtodbcx</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, UNIX, and Linux to test and troubleshoot connectivity to databases, create and execute SQL commands through ODBC, and run scripts.
Prerequisites

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, point to Programs, then MicroStrategy Tools, and then choose DB Query Tool.
   - On Windows from the command line, perform the following steps:
     - From the Windows Start menu, select Run. The Run dialog box opens.
     - In the Open drop-down list, type cmd and click OK. A command prompt opens.
     - Type todbcx.exe and press 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 11, Configuring MicroStrategy Using Command Line Tools.
   - On UNIX or Linux using the DB Query Tool interface, perform the following steps:
     - In a UNIX/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 ./mstrdbquerytool, then press ENTER.
   - On UNIX or Linux from the command line, perform the following steps:
     - In a UNIX/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 ./mstrtodbcx, then press 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 11, 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** to close the MicroStrategy DB Query Tool.

The DB Query Tool includes many useful features not discussed here. Refer to the **DB Query Tool Online Help** for details.

## 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 or UNIX machine using a `response.ini` file:

- **Creating a response.ini file**, page 358
• *Using the response.ini file to configure MicroStrategy, page 371*

• *Parameters and options in the response.ini file, page 371*

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 164.*

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 141.*

### 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 UNIX and Linux machines.

---

**To configure MicroStrategy using the command line mode**

1. In a UNIX or Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the directory you specified as the Home Directory during installation.

2. Browse to the `bin` directory.

3. At the command prompt, type `mstrcfgwiz-editor`, then press ENTER. The Configuration Wizard opens in command line mode.

   The sections or pages of the wizard displayed depend on your selections.

4. Review the information on the welcome screen and press ENTER to continue.

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 press 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 371.*

   • Type 2, and then press ENTER to create a new response.ini file. You can select from various configuration tasks, as described in the *Configuration tasks, page 358* section within this procedure.

---

**Configuration tasks**

6. You can support the configuration tasks described in the sections listed below:

   • Type 1, and then press ENTER to create metadata, History List, and statistics tables. Refer to *Creating metadata, History List, and statistics tables, page 359* for steps to create metadata and statistics tables.
• Type 2, and then press ENTER to configure a MicroStrategy Intelligence Server definition. Refer to Setting up MicroStrategy Intelligence Server, page 366 for steps to configure an Intelligence Server definition.

• Type 3, and then press ENTER to create project sources. Refer to Creating a project source, page 370 for steps to create project sources.

Creating metadata, History List, and statistics tables

If you selected option 1 in Configuration tasks, page 358, you can create metadata tables, History List tables, and statistics tables. The steps to perform these configuration tasks are provided separately in the sections below:

• Creating metadata tables, page 359
• Creating History List tables, page 361
• Creating statistics tables, page 364

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.

Prerequisite

• 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 358.

To create metadata tables

1 In the prompt asking whether to create metadata tables, type Y, and then press 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 press ENTER.

   If you do not have a DSN defined on your UNIX or Linux machine, see Creating a DSN for a data source, page 354.

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 Y and press enter, all information in the existing metadata repository is overwritten when the response file is executed at a later time.
4 Depending on your database type, you may be prompted to provide a login and password to your DSN:
   a  Type a login name for your database that stores your metadata tables, and then press ENTER. You are then prompted to provide a password for the login name.
   b  Type a password for the login name provided, and then press ENTER. You are then prompted to provide a metadata prefix for the metadata tables.

5 Depending on your database type, 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 press ENTER.
   •  Leave the prompt blank, and then press ENTER to provide no metadata prefix.

6 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 360 section within this procedure.
   •  If the data source does not point to a DB2 MVS database, the step to select a metadata script is displayed. This step is described in the To select a metadata script, page 360 section within this procedure.

To configure DB2 MVS database options

These steps are displayed if you are creating your metadata tables in a DB2 MVS database.

7 You can enter the database name to use or use the default name, as described below:
   •  Type the database name to use, and then press ENTER.
   •  Leave the prompt blank, and then press ENTER to use the default database.

You are then prompted to provide the MVS table space name.

8 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 press ENTER.
   •  Leave the prompt blank, and then press ENTER to use the default table space name.

You are then prompted to select a metadata script to create the metadata tables.

To select a metadata script

9 You can select the script used to create the metadata tables or use the default script, as described below:
   •  Enter a valid path to a script file, and then press ENTER.
• Leave the field blank, and then press ENTER to use the default script for your database type.

You are then prompted to create History List tables.

To create History List tables

10 You can choose whether to create History List tables or not, as described below:

• Type Y, and then press ENTER to create History List tables. Creating History List tables is described in Creating statistics tables, page 364.
• Type N, and then press ENTER to skip History List table creation. You are then prompted to create statistics tables, as described in the To create statistics tables, page 361 section within this procedure.

To create statistics tables

11 You can choose whether to create statistics tables or not, as described below.

• Type Y, and then press ENTER to create statistics tables. Creating statistics tables is described in Creating statistics tables, page 364.
• Type N, and then press 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 361 section within this procedure.

To generate a response.ini file

12 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 press ENTER. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

13 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 press 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 371.

• Type N, and then press 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 371.

Creating History List tables

You can create History List tables in a data source, as described in the procedure below.
Prerequisite

- 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 358*.

To create History List tables

1. After you create metadata tables (see *Creating metadata tables, page 359*) 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 press ENTER. You are then prompted for ODBC data source information.

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 362* 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 press 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 363* section within this procedure.
     
     — Type N, and then press ENTER to provide different ODBC DSN information, which is described in the *To provide ODBC DSN information, page 362* section within this procedure.

To provide ODBC DSN information

3. Type the number corresponding to the ODBC DSN for a database to create your History List tables in, and then press ENTER.

   ![Important](If you do not have a DSN defined on your UNIX or Linux machine, see *Creating a DSN for a data source, page 354*.)

4. Depending on your database type, you may be prompted to provide a login and password to your DSN:
   
   a. Type a login name for your database to create your History List tables in, and then press ENTER. You are then prompted to provide a password for the login name.
   
   b. Type a password for the login name provided, and then press 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 363 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 History List script, page 363 section within this procedure.

To configure DB2 MVS database options

These steps are displayed if you are creating your metadata tables in a DB2 MVS database.

You can enter the database name to use or use the default name, as described below:

- Type the database name to use, and then press ENTER.

- Leave the prompt blank, and then press ENTER to use the default database.

You are then prompted to provide the MVS table space name.

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 press ENTER.

- Leave the prompt blank, and then press 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 press ENTER.

- Leave the field blank, and then press ENTER to use the default script for your database type.

You are then prompted to create statistics tables.

To create statistics tables

You can choose whether to create statistics tables or not, as described below.

- Type Y, and then press ENTER to create statistics tables. Creating statistics tables is described in Creating statistics tables, page 364.

- Type N, and then press 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 364 section within this procedure.
To generate a response.ini file

10 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 press ENTER. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

11 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 press 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 371.

- Type N, and then press 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 371.

Creating statistics tables

You can create statistics tables in a data source, as described in the procedure below.

Prerequisites

- 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 358.

To create statistics tables

1 After you create metadata tables (see Creating metadata tables, page 359), create History List tables (Creating History List tables, page 361), or skip both of these procedures, you are prompted to create statistics tables.

To create statistics tables, type Y, and then press ENTER. You are then prompted for ODBC data source information.

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 365 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 press 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 366* section within this procedure.

— Type N, and then press ENTER to provide different ODBC DSN information, which is described in the *To provide ODBC DSN information, page 365* section within this procedure.

**To provide ODBC DSN information**

3 Type a valid ODBC DSN for a database to create your statistics tables in, and then press ENTER. You are then prompted to provide a login to your DSN.

4 Type a login name for your database to create your statistics tables in, and then press ENTER. You are then prompted to provide a password for the login name.

5 Type a password for the login name provided, and then press 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 365* 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 366* section within this procedure.

**To configure DB2 MVS database options**

These steps are displayed if you are creating your metadata tables in a DB2 MVS database.

6 You can enter the database name to use or use the default name, as described below:

- Type the database name to use, and then press ENTER.
- Leave the prompt blank, and then press ENTER to use the default database.

You are then prompted to provide the MVS table space name.

7 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 press ENTER.
- Leave the prompt blank, and then press 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

8 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 press ENTER.
- Leave the field blank, and then press 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

9 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 press ENTER. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

10 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 press 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 371.
- Type N, and then press 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 371.

Setting up MicroStrategy Intelligence Server

If you selected option 2 in Configuration tasks, page 358, 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 press 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 press ENTER. You are then prompted to provide a password for the login name.

3 Type a password for the login name provided, and then press ENTER. You are then prompted to provide a metadata prefix.
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 press ENTER.
- Leave the prompt blank, and then press ENTER to provide no metadata prefix.

You are then prompted to provide a temp table prefix.

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 press ENTER.
- Leave the prompt blank, and then press ENTER to provide no temp table prefix.

You are then prompted to provide a MicroStrategy user login.

Type a valid MicroStrategy user login that has administrator privileges, and then press 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 your initial configuration.

Type a password for the MicroStrategy user login provided, and then press ENTER. You are then prompted to choose the type of Intelligence Server configuration to complete.

You can perform one of the Intelligence Server configuration tasks, which are described in the sections below:

- Type 1, and then press ENTER to create a new server definition. Refer to Creating and using a server definition, page 367 for steps to create a new server definition.
- Type 2, and then press 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 367.
- Type 3, and then press ENTER to delete a server definition. Refer to Deleting a server definition, page 369 for steps to delete a server definition.
- Type 4, and then press 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 367.

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 366.*

**Prerequisites**

- 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 366.*

**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 press ENTER. You can press 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 press 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 press 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 press ENTER to use the default port number.
   - Type a port number, and then press 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 press ENTER to register Intelligence Server as a service. To perform this configuration, you must be logged into your UNIX or Linux machine with an account that has root level access and permissions.
   - Type N, and then press ENTER to not register Intelligence Server as a service.

   You are then prompted whether to start Intelligence Server when finished.

6. Type Y and press 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**

7. 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 press ENTER. The response.ini file is generated, and you are prompted whether to run the configuration immediately.
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 press 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 371.

- Type N, and then press 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 371.

**Deleting a server definition**

You can delete a server definition to remove it from the available server definitions for Intelligence Server.

**Prerequisites**

- 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 366.

**To delete a server definition**

1. In the prompt that asks for server definitions to be removed, type the name that distinguishes the server definition, and press 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.

**To generate a response.ini file**

2. 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 press ENTER. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

3. 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 press 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 371.

- Type N, and then press 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 371.
**Creating a project source**

If you selected option 3 in *Configuration tasks, page 358*, 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 press `ENTER`. You are then prompted to provide the Intelligence Server name.

2. Type the Intelligence Server name, and then press `ENTER`. You can also press `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 press `ENTER` to use the default port number.
   - Type a port number, and then press `ENTER`.

You are then prompted to define a time interval for a project source connection time out.

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 press `ENTER`.

You are then prompted to select an authentication type for the project source.

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 163*.

You are then prompted to provide a name for the `response.ini` file.

---

**To generate a response.ini file**

6. 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 press `ENTER`. The `response.ini` file is generated, and you are prompted whether to run the configuration immediately.

7. 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 press `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 371.*
Type N, and then press 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 371.

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 358.

To use the response.ini file through the Configuration Wizard in command line mode

1 In a UNIX or Linux console window, browse to HOME_PATH where HOME_PATH is the directory you specified as the Home Directory during installation.

2 Browse to the folder bin.

3 Type mstrcfgwiz-editor -r ReponseFile, where ReponseFile is the full qualified path to the response.ini file. For example:

   mstrcfgwiz-editor -r 
   /home/username/MicroStrategy/RESPONSE.INI

4 Press 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 167.

Configuring and controlling Intelligence Server

MicroStrategy provides various command line tools to configure and control Intelligence Servers running on UNIX and 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 UNIX or 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

1. From a UNIX/Linux console window, browse to HOME_PATH, where HOME_PATH is the directory that you specified as the home directory during installation.

2. Browse to the folder bin.

3. Type mstrsvr, and then press ENTER to start Intelligence Server and display available configuration and monitoring options.

4. 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.

5. To quit the tool and stop Intelligence Server, type S, and then press ENTER.
Configuring the default server instance with mstrsvr-configure

You can configure the default server instance for Intelligence Server using 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 UNIX/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 press ENTER. You are then prompted to provide a port number for Intelligence Server.

4. Type a port number, and then press ENTER. You are then prompted to provide a DSN to connect to.

5. Type a DSN, and then press ENTER. You are then prompted to provide a valid login for the DSN.

6. Type a valid login for the DSN, and then press ENTER. You are then prompted to provide a password for the DSN login.

7. Type a valid password for the DSN login, and then press ENTER. You are then prompted to provide a server definition name.

8. Type a server definition name, and then press ENTER. Your default server instance is configured.

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 UNIX/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 press ENTER. Help information is displayed, which provides syntax standards and available configuration options.

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 374.

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.

Be aware of the following:

- 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 UNIX or 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 UNIX or Linux operating system console, you must make sure reserved words and characters are not mistakenly included in your commands.

UNIX and 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 UNIX and Linux, page 50* in *Chapter 1, Planning Your Installation*, for an installation and configuration checklist.
ADDINU OR REMOVING MICROSTRATEGY COMPONENTS

This chapter describes how to add or remove MicroStrategy components on different operating systems.

This chapter includes the following sections:

- Adding or removing MicroStrategy components on Windows ........................................377
- Re-installing MicroStrategy components on Windows ..................................................378
- Uninstalling MicroStrategy components on Windows ..................................................380
- Uninstalling MicroStrategy components on UNIX and Linux ..................................381

Adding or removing MicroStrategy components on Windows

You can choose to add or remove one or more MicroStrategy components.

If you installed the MicroStrategy components using a disk, you need your original installation disk to add or remove MicroStrategy components.

To add or remove MicroStrategy components

1 Close all installed MicroStrategy products.

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.

3 Within the list of installed programs, select **MicroStrategy** and click **Change**. The MicroStrategy Setup Maintenance program opens.
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 selecting their check boxes. Clear the check boxes for the components you want to uninstall. 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 you are prompted to stop your Web server, click **Yes** to stop it and continue with adding or removing files.

9 Verify the settings and click **Next** to begin copying or removing the files.

10 After the modification routine is complete, click **Finish** to close the maintenance program. To fully remove MicroStrategy Office or MicroStrategy Health Center, see *Uninstalling MicroStrategy Office, page 381* and *Uninstalling MicroStrategy Health Center, page 381*.

   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 you face 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**

1 Close all installed MicroStrategy products.

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.
3 Within the list of installed programs, select **MicroStrategy** and click **Change**. The MicroStrategy Setup Maintenance program opens.

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 you are prompted to stop your Web server, click **Yes** to stop it and continue with the re-installation.

8 As part of a repair installation, you can also designate this machine as a Health Agent. Provide this configuration information as required and continue with the re-installation routine.

9 After the re-installation routine is complete, click **Finish** to close the maintenance program.

For details on each page of the MicroStrategy Installation Wizard, see Chapter 2, *Installing MicroStrategy on Windows*.

**Re-installing MicroStrategy Office**

This section describes the re-installation procedure for MicroStrategy Office.

---

**To re-install MicroStrategy Office**

1 Close all installed MicroStrategy products.

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.

3 Within the list of installed programs, select **MicroStrategy Office** and click **Change**. The MicroStrategy Office Setup Maintenance program opens.

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 you are 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** to close the maintenance program.

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 Uninst.isu 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.

To uninstall MicroStrategy components on Windows

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 Remove and click Next.
5. Click Yes to any prompts that appear.
6. If you are prompted to stop your Web server, click Yes to stop it and continue with the uninstallation.
7. After the uninstall routine is complete, select Yes to restart your computer, or No to restart it later.
8. Click Finish to close the maintenance program.

You should restart the computer for a clean uninstall.
Uninstalling MicroStrategy Office

This section describes the steps to uninstall MicroStrategy Office.

To uninstall MicroStrategy Office

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.
3. Within the list of installed programs, select **MicroStrategy Office** and click **Change**. The MicroStrategy Office Setup Maintenance program opens.
4. Select **Remove** and click **Next**.
5. You are prompted to select **Yes** to continue with the uninstallation procedure.
6. After the uninstallation routine is complete, click **Finish** to close the maintenance program.

Uninstalling MicroStrategy Health Center

This section describes the steps to uninstall MicroStrategy Health Center. These steps are required to remove all supporting files for Health Center.

To uninstall MicroStrategy Health Center

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.
3. Within the list of installed programs, select **Health Center** and click **Change**. The MicroStrategy Health Center Setup Maintenance program opens.
4. Select **Remove** and click **Start**.
5. After the uninstallation routine is complete, click **Finish** to close the maintenance program.

Uninstalling MicroStrategy components on UNIX and Linux

This section discusses how to uninstall MicroStrategy Intelligence Server and other MicroStrategy products on a UNIX or Linux platform.
To uninstall MicroStrategy products on UNIX and Linux

1. In a UNIX or 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 press ENTER. Follow the steps described in To complete uninstallation with the MicroStrategy Installation Wizard, page 382 below to complete the uninstallation process.

   • To uninstall through the command line, type ./setup.sh -console and press ENTER. Follow the command line prompts to complete the uninstallation process.

To complete uninstallation with the MicroStrategy Installation Wizard

4. A dialog box opens, prompting you to select 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 and continue with the steps To specify which MicroStrategy products to uninstall, page 382 below.

   • Uninstall: Select this option to uninstall all MicroStrategy products installed on the machine. Click Next and continue with the steps To specify which MicroStrategy products to uninstall, page 382 below.

To specify which MicroStrategy products to uninstall

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.

To complete the uninstallation process

12 On the Start installer operation page, verify the information and click Start. The products listed are uninstalled.

13 After uninstallation is complete, a message is displayed. Click Finish to complete the process and exit the MicroStrategy Installation Wizard.
EXPORT ENGINE CONFIGURATION

The following sections include steps for further configuration of the Export Engine:

Installation of Export Engine

The Export Engine will be installed automatically by the Intelligence Server installer. The PDF Exporter Service will use port 20100 by default.

Windows installations

For Windows installations, the service will be started automatically after Intelligence Server installation or each time the server is restarted. You find and control this service through Windows Task Manager or Windows services panel, listed as MSTR_PDFExporter.

Linux installations

Linux installations require the Export Engine to be started manually after installation or restart. 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.

Changing the port of the Export Engine

The PDF Exporter Service is set to use port 20100 by default.
**Steps to change the port used by the PDF Exporter Service**

1. Edit the `application.properties` file located under `{path to intelligence server}/PDFExporterService`
2. Change the value for `server.port` to the desired port number.
3. Restart the PDF Exporter Service.
4. Change the Intelligence Server setting to connect to the new location of the PDF Exporter Service. See, *Connecting Intelligence Server to a specific Export Engine* for steps to complete this process.

**Connecting Intelligence Server to a specific Export Engine**

You can configure each Intelligence Server to use a specific instance of the PDF Exporter Service.

**Steps to connect an Intelligence Server to a specific PDF Exporter Service**

1. Open the registry file.
2. Define the PDF Exporter Service host.
   
   "[HKEY_LOCAL_MACHINE\SOFTWARE\MicroStrategy\DSS Server\PDFExporter]":
   
   "host"="localhost"
3. Define the PDF Exporter Service port number.
   
   "[HKEY_LOCAL_MACHINE\SOFTWARE\MicroStrategy\DSS Server\PDFExporter]":
   
   "port"=dword:00004E84
4. Restart your Intelligence Server.

**Connect the Export Engine to a specific Kafka server**

You can utilize the enhanced logging abilities provided by Kafka by connecting your PDF Exporter Service to a specific Kafka server.
Steps to connect to a specific Kafka Server

1. Open the application.properties file under {path to Intelligence Server}/PDFExporterService
2. Define the Kafka server host: logging.kafka.server = 10.10.10.10
3. Define the Kafka server port: logging.kafka.port = 9093
4. Restart the PDF Exporter Service.

Increase concurrency of Export Engine

Follow these steps to increase the concurrency of the PDF Exporter Service:

1. Edit the application.properties file under path {path of iserver}/PDFExporterService/
2. Increase exporter.phantomjs.process.max value
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 UNIX and Linux, you can also configure parameters with the odbc.ini file. This appendix discusses the following topics:

- **Creating DSNs for specific data sources, page 387**: Configuration information required to create a DSN for data sources available through the Connectivity Wizard.
- **Creating database connections in Web, page 422**: 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 424**: Configuration information required to configure data sources using the odbc.ini file in a UNIX or 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 41*.

- 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 UNIX or 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>Driver Details</th>
<th>Details</th>
</tr>
</thead>
</table>
| **MicroStrategy ODBC Driver for Apache Hive Wire Protocol for Windows and UNIX/Linux, page 391** | • Data source name  
• Host name  
• Database name  
• Port number (10000 in most cases) |
| **MicroStrategy ODBC Driver for Amazon Redshift Wire Protocol for Windows and UNIX/Linux, page 391** | • Data source name  
• Host name  
• Port number (5439 in most cases)  
• Database name |
| **MicroStrategy ODBC Driver for DB2 Wire Protocol for Windows and UNIX/Linux, page 392** | • Data source name  
• Host name  
• Database name  
• Port number |
| **MicroStrategy ODBC Driver for DB2 Wire Protocol for iSeries for Windows and UNIX/Linux, page 392** | • Data source name  
• IP address  
• Collection  
• Location  
• Isolation level  
• Package owner  
• TCP port (446 in most cases) |
| **MicroStrategy ODBC Driver for DB2 z/OS for Windows and UNIX/Linux, page 393** | • Data source name  
• IP address  
• Collection  
• Location  
• Package collection  
• Package owner  
• TCP port (446 in most cases) |
| **MicroStrategy ODBC Driver for Impala Wire Protocol for Windows and UNIX/Linux, page 393** | • Data source name  
• Host name  
• Port number (21050 in most cases)  
• Database name |
<p>| <strong>MicroStrategy ODBC Driver for Informix Wire Protocol for</strong> | • Data source name |</p>
<table>
<thead>
<tr>
<th>Driver Details</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Windows and UNIX/Linux, page 394</strong></td>
<td>• Server name</td>
</tr>
<tr>
<td></td>
<td>• Host name</td>
</tr>
<tr>
<td></td>
<td>• Port number (1526 in most cases)</td>
</tr>
<tr>
<td></td>
<td>• Database name</td>
</tr>
<tr>
<td><strong>MicroStrategy ODBC Driver for Microsoft SQL Server for Windows and UNIX/Linux, page 394</strong></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>
<tr>
<td><strong>MicroStrategy ODBC Driver for MongoDB for Windows and UNIX/Linux, page 395</strong></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><strong>MicroStrategy ODBC Driver for Oracle Wire Protocol for Windows and UNIX/Linux, page 396</strong></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>
<tr>
<td></td>
<td>• TNSNames file</td>
</tr>
<tr>
<td><strong>MicroStrategy ODBC Driver for Pivotal Greenplum Wire Protocol for Windows and UNIX/Linux, page 397</strong></td>
<td>• Data source name</td>
</tr>
<tr>
<td></td>
<td>• Host name</td>
</tr>
<tr>
<td></td>
<td>• Port number (5432 in most cases)</td>
</tr>
<tr>
<td></td>
<td>• Database name</td>
</tr>
<tr>
<td>Driver Details</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>---</td>
</tr>
<tr>
<td><strong>MicroStrategy ODBC Driver for PostgreSQL Wire Protocol for Windows and UNIX/Linux, page 397</strong></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>• User Name</td>
</tr>
<tr>
<td><strong>MicroStrategy ODBC Driver for SAP Sybase ASE Wire Protocol for Windows and UNIX/Linux, page 397</strong></td>
<td>• Data source name</td>
</tr>
<tr>
<td></td>
<td>• Network address</td>
</tr>
<tr>
<td></td>
<td>• Database name</td>
</tr>
<tr>
<td></td>
<td>• Enable unicode support</td>
</tr>
<tr>
<td><strong>MicroStrategy ODBC Driver for SequeLink, page 398</strong></td>
<td>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 UNIX or Linux machine.</td>
</tr>
<tr>
<td><strong>MicroStrategy ODBC Driver for Salesforce, page 402</strong></td>
<td>The MicroStrategy ODBC Driver for Salesforce allows you to access resources on Salesforce.com, from an Intelligence Server hosted on a UNIX or Linux machine.</td>
</tr>
</tbody>
</table>

This section also provides information on how to install and configure drivers from other vendors with MicroStrategy:

- **ODBC Driver for Red Brick for UNIX/Linux, page 402**
- **ODBC Driver for Sybase Adaptive Server IQ for UNIX/Linux, page 403**
- **ODBC Driver for Teradata for UNIX/Linux, page 405**
- **ODBC Driver for Informix 8 for UNIX, page 406**
- **ODBC Driver for Netezza for UNIX/Linux, page 407**
- **ODBC Driver for MySQL 5.x for Linux, page 409**
- **ODBC Driver for Aster Database for Linux, page 410**
- **ODBC Driver for DataDirect Cloud for Linux, page 412**
- **ODBC Driver for Amazon Redshift for Linux, page 413**
- **ODBC Driver for Vertica for Linux or Solaris, page 415**
- **ODBC Driver for SAP HANA for Windows and Linux, page 416**
- **Other data sources and relational databases for Windows, page 420**
MicroStrategy ODBC Driver for Apache Hive Wire Protocol for Windows and UNIX/Linux

The following information is required for setting up the driver connection for the MicroStrategy ODBC Driver for Apache Hive Wire Protocol:

- **Data Source Name**: A name to identify the Apache Hive data source configuration in MicroStrategy. For example, Finance or ApacheHive-1 can serve to identify the connection.

- **Host Name**: The name or IP address of the machine on which the Apache Hive data source resides. The system administrator or database administrator assigns the host name.

- **Database Name**: The name of the database to connect to by default. If no database name is provided, the default database is used for the connection. The database administrator assigns the database name.

- **Port Number**: The port number for the connection. The default port number for Apache Hive is usually 20000. Check with your database administrator for the correct number.

- **Use Native Catalog Functions**: By default, native catalog functions are used to retrieve catalog information, rather than using ODBC catalog functions. This commonly improves performance of this type of data retrieval. However, this is only supported for HiveServer2 instances. Clear this check box if your data source connection supports the use of HiveServer1 instances.

MicroStrategy ODBC Driver for Amazon Redshift Wire Protocol for Windows and UNIX/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 UNIX/Linux

The following information is required for setting up the driver connection for MicroStrategy ODBC Driver for DB2 when running against 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 you should check with your database administrator for the correct number.

MicroStrategy ODBC Driver for DB2 Wire Protocol for iSeries for Windows and UNIX/Linux

The following information is required for setting up the driver connection for MicroStrategy ODBC Driver for DB2 Wire Protocol for iSeries/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 DB2 z/OS for Windows and UNIX/Linux

The following information is required for setting up the driver connection for MicroStrategy ODBC Driver for DB2 z/OS (formerly known as OS/390):

- **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 Impala Wire Protocol for Windows and UNIX/Linux

The following information is required for setting up the driver connection for the MicroStrategy ODBC Driver for Impala Wire Protocol:

- **Data Source Name**: A name to identify the Impala data source configuration in MicroStrategy. For example, Finance or Impala-1 can serve to identify the connection.

- **Host Name**: The name or IP address of the machine on which the Impala data source resides. The system administrator or database administrator assigns the host name.

- **Port Number**: The port number for the connection. The default port number for Impala is usually **21050**. Check with your database administrator for the correct number.
• **Database Name**: The name of the database to connect to by default. If no database name is provided, the default database is used for the connection. The database administrator assigns the database name.

**MicroStrategy ODBC Driver for Informix Wire Protocol for Windows and UNIX/Linux**

The following information is required for setting up the driver connection for the MicroStrategy ODBC Driver for Informix Wire Protocol:

• **Data Source Name**: A name to identify the Informix data source configuration in MicroStrategy. For example, Finance or Informix-1 can serve to identify the connection.

• **Server Name**: The client connection string designating the server and database to be accessed.

• **Host Name**: The name of the machine on which the Informix server resides. The system administrator or database administrator assigns the host name.

• **Port Number**: The Informix server listener's port number. The default port number for Informix is commonly 1526.

• **Database Name**: The name of the database to connect to by default, which is assigned by the database administrator.

**MicroStrategy ODBC Driver for Microsoft SQL Server for Windows and UNIX/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 SQL Database (Azure) support**: Defines whether the DSN is created to support SQL Azure. Select this check box if the DSN is used to access a SQL Azure data source.

---

**MicroStrategy ODBC Driver for MongoDB for Windows and UNIX/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 UNIX/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. This can be a server name such as Oracle-1 or an IP address such as 123.456.789.98.
  - **Port Number**: The Oracle listener port number provided by your database administrator. The default port number is usually 1521.
  - One of the following parameters; which one you choose is up to your personal preference:
    - **SID**: The Oracle System Identifier for the instance of Oracle running on the server. The default SID is usually ORCL.
    - **Service Name**: The global database name, which includes the database name and the domain name. For example, if your database name is finance and its domain is business.com the service name is finance.business.com.
  - **Alternate Servers**: A list of alternate database servers to enable connection failover for the driver. If the primary database server entered as the SID or service name is unavailable, a connection to the servers in this list is attempted until a connection can be established. You can list the servers in SID or service name format, as shown in the following examples:
    - Using an SID: (HostName=DB_server_name: PortNumber=1526:SID=ORCL)
    - Using a Service Name: (HostName=DB_server_name: PortNumber=1526:ServiceName=service.name.com)

- **TNSNames Connection**: A TNSNames connection uses a TNSNAMES.ORA file to retrieve host, port number, and SID information from a server (alias or Oracle net service name) listed in the TNSNAMES.ORA file. A TNSNames connection requires the following parameters:
  - **Server Name**: A server name, which is included in a TNSNAMES.ORA file included in the TNSNames File 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.
MicroStrategy ODBC Driver for Pivotal Greenplum Wire Protocol for Windows and UNIX/Linux

The following information is required for setting up the driver connection for the MicroStrategy ODBC Driver for Pivotal Greenplum Wire Protocol:

• **Data Source Name**: A name to identify the Greenplum data source configuration in MicroStrategy. For example, Finance or Greenplum-1 can serve to identify the connection.

• **Host Name**: The name or IP address of the machine on which the Greenplum data source resides. The system administrator or database administrator assigns the host name.

• **Port Number**: The port number for the connection. The default port number for Greenplum is usually **5432**. Check with your database administrator for the correct number.

• **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.

MicroStrategy ODBC Driver for PostgreSQL Wire Protocol for Windows and UNIX/Linux

The following information is required for setting up the driver connection for the MicroStrategy ODBC driver for PostgreSQL Wire Protocol:

• **Data Source Name**: A name to identify the PostgreSQL data source configuration in MicroStrategy. For example, Finance or PostgreSQL-1 can serve to identify the connection.

• **Host Name**: The name or IP address of the machine on which the PostgreSQL database resides. The system administrator or database administrator assigns the host name.

• **Port Number**: The port number for the connection. The default port number for PostgreSQL is usually **5432**. Check with your database administrator for the correct number.

• **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.

• **User Name**: The name of a valid user for the PostgreSQL database.

MicroStrategy ODBC Driver for SAP Sybase ASE Wire Protocol for Windows and UNIX/Linux

The following information is required for setting up the driver connection for the MicroStrategy ODBC driver for SAP Sybase ASE Wire Protocol:
• **Data Source Name**: A name to identify the Sybase ASE data source configuration in MicroStrategy. For example, Finance or SybaseASE-1 can serve to identify the connection.

• **Network Address**: The network address, in the format `ServerName_or_IPAddress,PortNumber`. For example, if your network supports named servers, you can specify an address such as `SybaseASE-1,5000`. You can also specify the IP address such as `123.456.789.98,5000`. Contact your system administrator for the server name or IP address.

• **Database Name**: The name of the database to connect to by default. The database administrator assigns the database name.

• **Enable Unicode support (UTF8)**: Select this check box if the database supports unicode.

**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 UNIX or 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, page 398*

• *Preparing the Microsoft Excel file, page 399*

• *Configuring the MicroStrategy ODBC driver for SequeLink, page 401*

**Preparing the Microsoft Access database**

You must complete the steps provided below to access an Access database stored on a Windows machine from an Intelligence Server hosted on a UNIX or Linux machine.

**Prerequisites**

• 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 136.*

**To access Microsoft Access databases from an Intelligence Server hosted on UNIX or Linux**

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 56*).
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.

To configure the SequeLink ODBC Socket Server

2 On the Windows machine where you installed the SequeLink ODBC Socket Server, from the Start menu, point to Programs, point to DataDirect SequeLink 5.5 Service for ODBC Socket, and then select SequeLink Management Console Snap-in.

3 Under Console Root, expand SequeLink 5.5 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 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 DSN=AccessDSN, where 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. The DataSourceLogonMethod Properties dialog box opens.

12 From the Value drop-down list, select Anonymous. This allows connection to the Access database 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 UNIX or 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 401.

Preparing the Microsoft Excel file

You must complete the steps provided below to access Excel files stored on a Windows machine from an Intelligence server hosted on a UNIX or Linux machine.
Prerequisites

- On 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 file, see Prepare an Excel file as a valid data source, page 421.
- 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 421.

To access Microsoft Excel files from an Intelligence Server hosted on UNIX or Linux

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 56).

To configure the SequeLink ODBC Socket Server

2 On the Windows machine where you installed the SequeLink ODBC Socket Server, from the Start menu, point to Programs, point to DataDirect SequeLink 5.5 Service for ODBC Socket, and then select SequeLink Management Console Snap-in.

3 Under Console Root, expand SequeLink 5.5 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 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 UNIX or 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 401.

**Configuring the MicroStrategy ODBC driver for SequeLink**

The steps provided 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.

---

**To configure the MicroStrategy ODBC driver for SequeLink**

1. On the UNIX or 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:

```
![Console Root]
  - Sequelink 5.5 Manager (localhost)
    - Connected to SLAgent55
      - 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 255.*

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 Red Brick for UNIX/Linux**

The ODBC driver for Red Brick is not a MicroStrategy-branded Linux. The following steps show how to configure ODBC driver for Red Brick.

---

**To create an ODBC Driver for Red Brick**

1. Install ODBC Driver for Red Brick for the correct operating system. For information on installation, refer to the *Installation and Configuration Guide for UNIX and Linux* provided by IBM.

    Make sure you install the Red Brick Client Products (version 6.2 and higher) so that they can be accessed by the appropriate users. You need the following components:

    - RISQL Entry Tool, RISQL Reporter, and Client TMU
    - Red Brick ODBCLib (SDK)
    - Red Brick ODBC Driver
The directory where Red Brick Client Products are installed should always be accessible to MicroStrategy Intelligence Server.

To define the location of the driver files

2 In a UNIX/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.

3 Add Write privileges to the ODBC.sh file by entering the following command:
   chmod u+w ODBC.sh

4 Edit the ODBC.sh file and provide the location of the directory where the Red Brick Client Products are installed. Within the ODBC.sh file, the following definition is included:

   RB_CONFIG='<RB_CONFIG>'

   Replace this <RB_CONFIG> placeholder with the location of where the Red Brick Client Products are installed. Do not modify any other occurrences of <RB_CONFIG> within odbc.sh.

5 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

6 In a UNIX/Linux console window, browse to HOME_PATH where HOME_PATH is the directory you specified as the Home Directory during installation.

7 Open the odbc.ini.example file and find the section that starts with [RED_BRICK_62] if you are using Red Brick 6.2 or [RED_BRICK_63] if you are using Red Brick 6.3. Copy the section into the odbc.ini file.

8 Edit the DSN parameters SERVER and DATABASE, and modify the value of RB_CONFIG with the location of the directory where the Red Brick Client Products are installed. For information on the available parameters, refer to your third-party Red Brick driver documentation. This can often be found along with the driver installation.

9 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 Sybase Adaptive Server IQ for UNIX/Linux

ODBC driver for Sybase Adaptive Server IQ is not a MicroStrategy-branded driver. The following steps show how to configure ODBC driver for Sybase Adaptive Server IQ.
To configure ODBC Driver for Sybase Adaptive Server IQ

1. Install ODBC Driver for Sybase Adaptive Server IQ for the correct operating system. For information on installation, refer to the Installation and Configuration Guide provided by Sybase.

   The directory where ODBC driver for Sybase Adaptive Server IQ is installed should always be accessible to MicroStrategy Intelligence Server.

To define the location of your environment

2. In a UNIX/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.

3. Add Write privileges to the ODBC.sh file by entering the following command:
   ```bash
   chmod u+w ODBC.sh
   ```

4. Edit the ODBC.sh file and add the location of the directory where the ODBC Driver for Sybase Adaptive Server IQ is installed. Within the ODBC.sh file, the following definitions are included:
   ```plaintext
   IQDIR15='<IQDIR15>'
   IQDIR16='<IQDIR16>'
   ```
   Replace the <IQDIR15> and <IQDIR16> placeholders, depending on your version of Sybase Adaptive Server IQ, with the directory path. Do not modify any other occurrences of <IQDIR15> or <IQDIR16> within odbc.sh.

5. 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

6. In a UNIX/Linux console window, browse to HOME_PATH where HOME_PATH is the directory you specified as the home directory during installation.

7. Open the odbc.ini.example file and find the section that starts with [SAP SYBASEIQVersion], where Version is the version of Sybase Adaptive Server IQ. Copy the section into the odbc.ini file.

8. Edit the DSN parameters EngineName, DatabaseName and CommLinks, and modify the value of ASDIR with the location of the directory where the ODBC Driver for Sybase Adaptive Server IQ is installed. For information on the available parameters, refer to your third-party Sybase driver documentation. This can often be found along with the driver installation.

For details on these DSN parameters, refer to the product documentation provided directly by the database vendor.

**ODBC Driver for Teradata for UNIX/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 420.

**To configure the ODBC Driver for Teradata**

1. 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.

   Note the following:
   
   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 specify the Teradata GSS library installation (AIX only)**

2. For AIX only, you need to specify the Teradata GSS library installation in the `ODBC.sh`. In a UNIX/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`.

3. Add Write privileges to the `ODBC.sh` file by entering the following command:
   
   ```
   chmod u+w ODBC.sh
   ```

4. Edit the `ODBC.sh` file and provide the location of the local Teradata GSS library installation. Within the `ODBC.sh` file, the following definition is included:
   
   ```
   MSTR_TERADATA_PATH='<MSTR_TERADATA_PATH>'
   ```
   
   Replace this `<MSTR_TERADATA_PATH>` placeholder with the directory path. Do not modify any other occurrences of `<MSTR_TERADATA_PATH>` within `odbc.sh`.

5. 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

6 In a UNIX/Linux console window, browse to HOME_PATH where HOME_PATH is the directory you specified as the Home Directory during installation.

7 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.

8 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.

9 Save the odbc.ini file.

ODBC Driver for Informix 8 for UNIX

The MicroStrategy ODBC Driver for Informix 8 is already installed in the INSTALL_PATH/lib directory.

The ODBC Driver for Informix 8 for UNIX and 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

1 Install the Informix CSDK.

   The directory where CSDK is installed should always be accessible to Intelligence Server.

To configure the environment

2 In a UNIX 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.

3 Add Write privileges to the ODBC.sh file by entering the following command:
4 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:
  
  ```
  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:
  
  ```
  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`.

5 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

6 In a UNIX console window, browse to `HOME_PATH` where `HOME_PATH` is the directory you specified as the home directory during installation.

7 Open the `odbc.ini.example` file and search for the section that starts with `[IBM INFORMIX]`. Copy the section into the `odbc.ini` file.

8 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.

9 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 UNIX/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

1. Install 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.

To modify the odbcinst.ini file

2. In a UNIX/Linux console window, browse to \textit{HOME\_PATH}, where \textit{HOME\_PATH} is the directory you specified as the home directory during installation.

3. Edit the \texttt{odbcinst.ini} file and replace all instances of \texttt{<NETEZZA\_ODBC\_DIR>\textit{}} 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:
   \begin{verbatim}
   Driver = /<NETEZZA\_ODBC\_DIR>/lib64/libnzodbc.so
   \end{verbatim}

   Then the modified path will be:
   \begin{verbatim}
   Driver = /usr/odbc/netezzahome/lib64/libnzodbc.so
   \end{verbatim}

4. Save the \texttt{odbcinst.ini} file.

To modify the odbc.ini file

5. Open the \texttt{odbc.ini.example} file and search for the section that starts with [\texttt{IBM\_NETEZZA}].

6. Open the MicroStrategy \texttt{odbc.ini} file.

7. Copy and paste the contents from the \texttt{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.

8. 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.

9. Save the \texttt{odbc.ini} file.

For details on these DSN parameters, refer to the product documentation provided by the database vendor.
ODBC Driver for MySQL 5.x for Linux

The ODBC driver for MySQL 5.x is not a MicroStrategy-branded driver. The following steps show how to configure the ODBC driver for MySQL 5.x, which is certified for the Linux operating system.

You must modify the odbc.ini file to create the DSN for MySQL 5.x.

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 5.x

1. Install the 64-bit ODBC Driver for MySQL for the Linux operating system, found at the hyperlink 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.

To define the location of the driver files

2. In a UNIX/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.

3. Add Write privileges to the ODBC.sh file by entering the following command:
   chmod u+w ODBC.sh

4. 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='&lt;MYSQL_PATH&gt;'

   Replace this &lt;MYSQL_PATH&gt; placeholder with the location of where the MySQL driver is installed. Do not modify any other occurrences of &lt;MYSQL_PATH&gt; within odbc.sh.

5. 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

6 In a Linux console window, browse to HOME_PATH, where HOME_PATH is the directory you specified as the home directory during installation.

7 Open the odbc.ini.example file and search for the section that starts with [MySQL].

8 Open the MicroStrategy odbc.ini file.

9 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.

10 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.

   ! Ensure that there is no white space between the equals sign (=) which separates the parameter and its value.

11 Save the odbc.ini file.

   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 Community Server 5.x. To create a database instance and database connection, see Creating a database instance, page 179 and Creating a database connection, page 181.

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.

   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

1 Install 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.

To configure the environment

2 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.

3 Add Write privileges to the ODBC.sh file by entering the following command:
   chmod u+w ODBC.sh

4 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.

5 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

6 In a Linux console window, browse to HOME_PATH, where HOME_PATH is the directory you specified as the home directory during installation.

7 Open the odbc.ini.example file and search for the section that starts with [Aster Database].

8 Open the MicroStrategy odbc.ini file.

9 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.

10 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.
11. Ensure that there is no white space between the equals sign (=) which separates the parameter and its value.

11. Save the odbc.ini file.

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

1. Install 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.

### To configure the environment

2. 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`.

3. Add Write privileges to the `ODBC.sh` file by entering the following command:

   ```
   chmod u+w ODBC.sh
   ```

4. 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.

5. 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

6 In a Linux console window, browse to HOME_PATH, where HOME_PATH is the directory you specified as the home directory during installation.

7 Open the odbc.ini.example file and search for the section that starts with [DataDirect Cloud].

8 Open the MicroStrategy odbc.ini file.

9 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.

10 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.

11 Save the odbc.ini file.

This completes the steps to create a DSN and configure an ODBC driver for DataDirect Cloud.

ODBC Driver for Amazon Redshift for Linux

The ODBC driver for Amazon Redshift is not a MicroStrategy-branded driver (for the MicroStrategy-branded driver, see MicroStrategy ODBC Driver for Amazon Redshift Wire Protocol for Windows and UNIX/Linux, page 391). The following steps show how to configure the ODBC driver for Amazon Redshift 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 Amazon Redshift

1 Install the Amazon Redshift 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 Amazon Redshift drivers certified with MicroStrategy, refer to the MicroStrategy General Information Readme.

**To configure the environment**

2 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`.

3 Add Write privileges to the `ODBC.sh` file by entering the following command:
   ```bash
cchmod u+w ODBC.sh
```

4 Edit the `ODBC.sh` file and make the following changes:
   • Provide the location of the directory where the Amazon Redshift ODBC driver files are installed. The following definition is included:
     ```bash
     REDSHIFT_PATH='<REDSHIFT_PATH>'
     ```
     Replace this `<REDSHIFT_PATH>` placeholder with the directory path. Do not modify any other occurrences of `<REDSHIFT_PATH>` within `odbc.sh`.
   • Provide the location of `amazon.redshiftodc.ini`. The following definition is included:
     ```bash
     AMAZONREDSHIFTODBCINI='<AMAZONREDSHIFTODBCINI>'
     ```
     Replace this `<AMAZONREDSHIFTODBCINI>` placeholder with the directory path. Do not modify any other occurrences of `<AMAZONREDSHIFTODBCINI>` within `odbc.sh`.

5 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**

6 In a Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory you specified as the home directory during installation.

7 Open the `odbc.ini.example` file and search for the section that starts with `[Amazon Redshift ODBC DSN 64]`.

8 Open the MicroStrategy `odbc.ini` file.

9 Copy and paste the contents from the `odbc.ini.example` file for your Amazon Redshift ODBC driver. You should paste the contents of the DSN exactly as they appear in the example file.

10 Make the following changes to the copied sample file:
    • Modify the driver location to match the location of the installed Amazon Redshift 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 Amazon Redshift 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.

11. Save the odbc.ini file.

This completes the steps to create a DSN and configure an ODBC driver for Amazon Redshift.

**ODBC Driver for Vertica for Linux or Solaris**

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 or Solaris.

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**

1. Install the ODBC Driver for Vertica for the Linux or Solaris operating system. For information on installation, refer to the product documentation provided by the database vendor.

   Note the following:
   - 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 Vertica drivers certified with MicroStrategy, refer to the MicroStrategy General Information Readme.

**To configure the environment**

2. 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.

3. Add Write privileges to the ODBC.sh file by entering the following command:

   ```bash
   chmod u+w ODBC.sh
   ```
4 Edit the ODBC.sh file and provide the location of the vertica.ini file. Within the ODBC.sh file, the following definition is included:

   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.

5 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

6 In a Linux or Solaris console window, browse to HOME_PATH, where HOME_PATH is the directory you specified as the home directory during installation.

7 Open the odbc.ini.example file and search for the section that starts with [HP VERTICA].

8 Open the MicroStrategy odbc.ini file.

9 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.

10 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.

11 Save the odbc.ini file.

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 1.x.

• To configure an ODBC Driver for SAP HANA on Windows, page 417
• To configure an ODBC Driver for SAP HANA on Linux, page 418
Prerequisites:

- 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 146 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.

   For exact version numbers of SAP HANA drivers certified with MicroStrategy, refer to the MicroStrategy General Information Readme.

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 137.

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 355.

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 179.

   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 419.
To configure an ODBC Driver for SAP HANA on Linux

1. Install 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.

   For exact version numbers of SAP HANA drivers certified with MicroStrategy, refer to the MicroStrategy General Information Readme.

To configure the environment

2. 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.

3. Add Write privileges to the ODBC.sh file by entering the following command:
   ```bash
   chmod u+w ODBC.sh
   ```

4. 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:
   ```bash
   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.

5. 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

6. In a Linux console window, browse to HOME_PATH where HOME_PATH is the directory you specified as the Home Directory during your MicroStrategy installation.

7. Open the odbc.ini.example file and find the section that starts with [SAP HANA]. Copy the section into the odbc.ini file.

8. 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 \textit{port} with the port number for the connection to the SAP HANA database. Contact your SAP HANA database administrator for the required port number.

- \textbf{USER=uid}
  - Replace \textit{uid} with a valid SAP HANA user account.

- \textbf{PASSWORD=pwd}
  - Replace \textit{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.

\textbf{9} Save the \texttt{odbc.ini} file.

\textbf{10} 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 \textit{Using the DB Query Tool, page 355}.

\textbf{11} 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 \textit{Creating a database instance, page 179}.

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 UNIX drivers.

This completes the steps for the initial connection to SAP HANA in MicroStrategy for Windows environments. For additional configuration requirements, see \textit{Additional requirements to support SAP HANA, page 419} below.

\textbf{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 \textit{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 \textit{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.

Teradata

If you intend to use Teradata, which is certified by MicroStrategy, you need to:

- Pre-install the NCR ODBC Driver for Teradata RDBMS.
- Ensure that Teradata DSNs are set to Run in Quiet Mode.

If you use Teradata, the following settings are required for setting up the driver connection.

In the Teradata ODBC Driver Options dialog box, click Options to set the following required options:

- **Session Mode**: Select Teradata as the session mode to apply for the duration of the session.
- **Date Time Format**: Set this value to AAA format so the ODBC driver handles the Dates, Time, and Timestamps as ANSI-compatible strings. The ANSI-compatible strings are only available with the V2R3 or later databases.
- **Disable Parsing**: Select this check box to disable parsing of SQL statements by the ODBC driver.

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*.

For information on other options, refer to the online help by clicking Help.

Microsoft Excel

A Microsoft Excel file can be used as a data source in MicroStrategy. The information provided 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.

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.

   You can create multiple tables in one Excel file by highlighting different parts of the file and assigning them different names.

3 Save the Excel file.
   - 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 179.

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 dashboards 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.

Prerequisite

• 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, click Create, then select Access External Data. Click 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. The Import from Tables page opens.

5 From the Data Sources panel, click Add. The Data Source dialog box opens.

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 387.*

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:

a  Click **Show connection string.**

b  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 387.*

c  Select the **Edit connection string** check box. You can now edit the connection string.

d  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.

e  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.

7  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** to create the connection. For more information on importing data into Web, see the *MicroStrategy Web Help.*
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 UNIX or Linux environment. Therefore this section is not relevant to ODBC and DSN connections on Windows.

For information on what operating systems each ODBC driver is certified for, see Certified ODBC drivers for MicroStrategy Intelligence Server, page 41.

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 UNIX or Linux. It contains the definitions for all of 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 UNIX or Linux. It is recommended copy the examples in the odbc.ini.example file to odbc.ini, to act as a basis for your configurations. This example file uses 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 following resources:

- For any data source listed in odbc.ini.example as a MicroStrategy driver, refer to DataDirect’s documentation at http://media.datadirect.com/download/docs/odbc/allodbc/help.html.
- For any data source that is not listed in odbc.ini.example as a MicroStrategy driver, refer to the documentation for the third-party vendor of the driver. This can often be found along with the driver installation.
Installing MicroStrategy Hadoop Gateway

MicroStrategy Hadoop Gateway is a data processing engine that can be installed in a Spark environment. The engine is a native connector that allows analysis of unstructured data in Hadoop and provides high-speed parallel data transfer between the Hadoop Distributed File System (HDFS) and MicroStrategy Intelligence Server. The MicroStrategy Hadoop Gateway is built on top of Spark technology and can take advantages on high performance of Spark.

MicroStrategy Hadoop Gateway consists of two parts:

• **Hadoop Gateway Connector** is a native connector. Hadoop Gateway Connector provides the ability to select one or more data files from Hadoop and load them into MicroStrategy Intelligence Server. Additionally, users can take advantage of data wrangling capabilities during data loading from Hadoop. After the data files are loaded, they are published as an in-memory data cube on Intelligence Server for report generation.

• **Gateway Manager** is a centralized management portal which enables administrators to create/modify/delete, deploy/un-deploy, and start/stop a Hadoop Connector. Additionally, the portal shows the status of each Gateway Connector and provides access to logs, for audit and troubleshooting purposes.

Connection Modes

MicroStrategy Hadoop Gateway supports the following deployment modes:

• **Local**
  - Spark driver runs on a single machine.
  - The number of threads simulates multiple worker nodes for the job.
  - Typically used for debug purposes..

• **Standalone**
• Spark service is configured as standalone mode on Hadoop cluster and exposes the Spark Master URL. Hadoop Gateway submits the job to Spark Master, which breaks down job to tasks and distributes to worker nodes.

• Spark Master takes care of resource management. Hadoop Gateway submits job to Spark Master for distribution manipulation.

• **YARN**
  - Spark Master is replaced by YARN for resource management and tasks distribution.
  - Supported by both Hortonworks and Cloudera clusters.
  - YARN service must be configured and started in a cluster. Additionally, the path of YARN dependency jar packages must be specified during configuration of MicroStrategy Hadoop Gateway.

**Constraints**

• Supported database files (files that can be selected and loaded into Intelligence Server): CSV (Comma Separated Values) and text.

• Though you can setup multiple connections to HDFS clusters in the management portal, only a single connection to an HDFS data source can exist at one time.

• Hortonworks does not support Hadoop Gateway Standalone deployment mode. This limitation is because Hortonworks does not include the Spark Master service.

• Supported authentication:
  - Shared Kerberos authentication is supported in YARN deployment mode.
  - Anonymous authentication is supported in Local and Standalone deployment mode.

**System requirements and supported configurations**

• Spark 1.6.x

• Supported distribution version:
  - CDH 5.7.0
  - HDP 2.5.0 and above

**Prerequisites**

• Hadoop environment is installed on UNIX/Linux servers
• Hadoop cluster is setup with at least one edge node

• A Linux machine that can deploy Hadoop Gateway (either a data node of the Spark cluster or any Linux machine that can communicate with the Spark cluster). JRE 7.0 or later must be installed.

• Knowledge of the following information for the Linux machine used to deploy MicroStrategy Hadoop Gateway:
  ▫ Secure Shell (SSH) port, username and password
  ▫ Host name or IP address
  ▫ Hadoop name node, HDFS port, and Web HDFS port

**Steps to deploy MicroStrategy Hadoop Gateway**

Perform one of the following procedures to install MicroStrategy Hadoop Gateway to your Spark cluster:

• [Deploy MicroStrategy Hadoop Gateway via Gateway Manager](#)

• [Deploy MicroStrategy Hadoop Gateway Manually](#)

**Deploy MicroStrategy Hadoop Gateway via Gateway Manager**

1. Login to MicroStrategy Web with administrator privilege and open a specific project.

2. Click **Add External Data**.
3 In the “Connect to Your Data” dialog box, place the mouse cursor over the **Hadoop** option and click **Browse Hadoop Files**.

4 In the “Connect to Hadoop” dialog box, click **Change Connection**...

5 In the “Data Source” dialog box, click **Manage Hadoop Gateway**.
MicroStrategy Web must be configured to use HTTPS and SSL certificates for the Manage Hadoop Gateway options to appear. For information about configuring HTTPS and SSL certificates, see the Installation and Configuration Guide.

The Gateway Manager appears and lists all managed Hadoop Gateways.

6 To add a Hadoop Gateway, click Add. The Hadoop Gateway dialog box appears.
In the “Hadoop Gateway” dialog box, enter the following information:

- **Name**: A unique name for the new Hadoop Gateway

**Hadoop Properties**

- **Hadoop NameNode**: Hadoop cluster name node server name or IP address
- **HDFS Port**: HDFS port number used to connect, default: 8020
- **WebHDFS Port**: Port number used by file access via HTTP protocol

**Gateway Properties**

- **Host**: Machine name or IP address used to install and deploy Hadoop Gateway.
- **Port**: Hadoop Gateway port number, default: 30004

**Spark Properties**

- **Mode**: Select one of the following Spark modes:
  - **Local**
    - **Hadoop NameNode**
    - **HDFS Port**: Port used to pull data from HDFS
    - **WebHDFS Port**: Port used to communicate with NameNode via HTTP protocol
    - **Host**: Host or IP address of server where Hadoop Gateway is deployed
    - **Port**: Port used by Hadoop Gateway
    - **Authentication**: Must be set to Anonymous
    - **Thread Number**: Number of threads to simulate
    - **Memory of driver**: Memory to use for driver process; 1g if left blank
- **Memory of executor**: Memory to use per executor process; 1g if left blank

- **Standalone**
  - **Hadoop Namenode**
  - **HDFS Port**: Port used to pull data from HDFS
  - **webHDFS Port**: Port used to communicate with NameNode via HTTP protocol
  - **Host**: Host or IP address of server where Hadoop Gateway is deployed
  - **Port**: Port used by Hadoop Gateway
  - **Authentication**: Must be set to Anonymous
  - **Master**: URL and port for Spark master
  - **Memory of driver**: Memory to use for driver process; 1g if left blank
  - **Memory of executor**: Memory to use per executor process; 1g if left blank
  - **CPU Cores**: Maximum CPU cores to request for the application from across the cluster

- **Yarn**
  Deploy Hadoop Gateway in YARN mode, the application is configured YARN - client mode, the driver runs in the client process, and the application master is only used for requesting resources from YARN. This option requires an enabled YARN service on the cluster.
  - **Hadoop Namenode**
  - **HDFS Port**: Port used to pull data from HDFS
  - **webHDFS Port**: Port used to communicate with NameNode via HTTP protocol
  - **Host**: Host or IP address of server where Hadoop Gateway is deployed
  - **Port**: Port used by Hadoop Gateway
  - **Authentication**:
    - **Jar of path**: "spark-assembly.jar" file path; obtain this information from cluster administrator.
    - **Memory of driver**: Memory to use for driver process; 1g if left blank
    - **Memory of executor**: Memory to use per executor process; 1g if left blank
  - **CPU Cores**: Maximum CPU cores to request for the application from across the cluster

  b. Click **Save**.

7 On the Gateway Manager, click an available action or option:

- **Deploy**: Deploys a Hadoop Gateway to the destination machine; the status changes to “Stopped”.
- **Start**: Starts a deployed Hadoop Gateway; the status changes to “Running”
- **Stop**: Stops a running Hadoop Gateway; the status changes to “Stopped”.
- **Undeploy**: Undeploys a deployed Hadoop Gateway; the status changes to “Undeployed”.
- **<name>**: Change certain properties or delete the Hadoop Gateway.
Deploy MicroStrategy Gateway Hadoop Manually

1. Download the MicroStrategy Hadoop Gateway package from the download site:
   a. Go to https://software.microstrategy.com/download/.
   b. Select the software version.
   c. Expand Add-Ons and Tools.
   d. Download the Hadoop tool. The Haddop Manager on Spark (HGoS) Manager package (hgos_manager-<version>.jar.tar.gz) is downloaded.

2. Upload the HGoS Manager package to the worker node where it will be installed. Place the package in the /opt/ directory.

3. Unzip the HGoS Manager package.
   For example: tar zxvf hgos-manager-0.1.0.jar.tar.gz

4. Edit the configuration file, spark.conf, located in the “conf” folder.
   • “spark.master” attribute: By default, the value is set to “local[2]”.
     ▪ Change “local” to the cluster master node address.
     ▪ Change “[2]” to the number of executors to be generated for a Spark job task.
   • Other fields as necessary. See the following examples for the supported modes.
     Standalone:

```
spark.cores.max=8
AuthMode=0
spark.executor.memory=5g
spark.master=spark://ash-109-33r.labs.microstrategy.com:7077
spark.files=/etc/hadoop/conf/hdfs-site.xml,/etc/hadoop/conf/core-site.xml
spark.driver.memory=5g
hgos.tcp.port=30005
spark.serializer=org.apache.spark.serializer.KryoSerializer
Mode=1
hgos.restful.port=4020
```

YARN:

```
spark.executor.memory=1g
hgos.tcp.port=30005
spark.driver.memory=1g
hgos.restful.port=4020
Mode=0
AuthMode=0
spark.master=local[3]
spark.cores.max=2
```
Local:

```
spark.executor.memory=lg
hgos.tcp.port=30005
spark.driver.memory=lg
hgos.restful.port=4020
Mode=0
AuthMode=0
spark.master=local[3]
spark.cores.max=2
```

5 Start the HGoS engine
   a Inside the HGoS Manager installation folder, execute the following shell script
      `/opt/hgos-0.1.0/sbin/start-hgos.sh`.

**Import Data from Hadoop**

For instructions to import data from Hadoop, see *Importing data from Hadoop* in the
Troubleshooting

This appendix provides information on common problems that you might encounter while installing and configuring MicroStrategy on UNIX, 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 UNIX 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

© 2017, MicroStrategy Inc.
INTELLIGENCE_SERVER_INSTALL_PATH\PDFGeneratorFiles. The default installation path for the Intelligence Server in UNIX is home\MicroStrategy\PDFGeneratorFiles. For the change to take effect, you must restart Intelligence Server.

Server port number errors

This section provides troubleshooting information on server port number errors.

I forgot the server port number

1. Open MicroStrategy Service Manager.
   - **Windows**: From the Start menu, point to Programs, then MicroStrategy Tools, and then select Service Manager. MicroStrategy Service Manager opens.
   - **UNIX/Linux**: In a UNIX or 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 bin and type ./mstrsvcman, then press ENTER. MicroStrategy Service Manager opens.
2. Click Options. This launches the Service Options dialog box.
3. Click Intelligence Server Options tab to view the port number.

Port number is in use

In a UNIX or 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 press ENTER.
3. Type mstrctl -s IntelligenceServer ci FindingPortNumber, then press ENTER.
4. Type mstrctl -s IntelligenceServer gs FindingPortNumber, then press 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 the window.
3. Enter the information required for each box and click Test. The connection is either successful, or an error message is displayed to explain the problem.

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 146.

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 UNIX or 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. In a console window, type:

   ```
   #chmod g-s directory
   
   where directory is the InstallPath for Intelligence Server.
   ```

2. Press ENTER.
GLOSSARY

A

account
Creates a relationship between access privileges and user login credentials. Account permissions are based on granted roles, and each role has specific privileges. See also: application administrator, application designer, subscription administrator.

Activation Code
A code used to activate MicroStrategy Intelligence Server after installation. This code is sent to an email address provided during activation.

ad hoc query
A SQL query dynamically constructed by desktop tools and whose results are not known before it is sent to the server. The user is asking a new question that has not been answered by an existing report.

address
Set of information that tells Narrowcast Server how to send services to a particular subscriber. Each address can be associated with one and only one login/user. Each address is defined to use a specific device. See also: device.

address display
A name that is displayed by receiving systems. For email delivery, the address display is used to identify the email address. For example, johnsmith@microstrategy-tutorial.demo might be displayed as “Smith, John” by the receiving email client.

address ID
An ID value that identifies individual addresses. Each address can be subscribed to a subscription set multiple times.

administration object
Narrowcast Server components that control the processing of Narrowcast messages, providing the means to acquire, format, and transmit messages to recipients. These objects are created and configured by the system administrator.

administrator
A user who installs and monitors software and user configurations, maintains the state of the software, and administers the MicroStrategy servers in the platform. An administrator also defines users, assigns user login accounts and user privileges, and analyzes the performance of the system.

aggregate data
Information or facts added together or "aggregated" to form summaries of information considered as a whole.
aggregate function
A numeric function that acts on a column of data and produces a single result. Examples include SUM, COUNT, MAX, MIN, and AVG.

aggregate table
A fact table that stores data that has been aggregated along one or more dimensions.

All Subscription Data
An XML string that returns subscription information. It requires you to select an XSL stylesheet to format the subscription information returned as desired.

analyst
A user who analyzes business data by accessing reports, performing drilling, and otherwise manipulating reports and documents to see required business data. An analyst receives useful data from information devices like smart phones and email without necessarily understanding how such information is derived or delivered.

analytical application
In MicroStrategy, a software application designed to provide predefined reports and other analytics based on a predefined metadata repository, for various industries to gain insight into their business data. The application is not fixed to a specific physical schema, giving it the flexibility to be ported to a company's existing data warehouse.

Analytical Engine
A component of the MicroStrategy Intelligence Server that performs all advanced analytical functions. The Analytical Engine evaluates functions not supported by the data warehouse RDBMS and it cross-tabulates reports.

analytics
Predefined tools that allow analysis within the Analysis Module's functional areas. Analytics include reports (graph, grid, and so on), scorecards, dashboards, and so on.

analytics library
The collection of reports and related objects in the MicroStrategy metadata repository. Library objects include reports, metrics, filters, and prompts. Library objects are defined based on attributes and facts (objects in the logical data model.)

application administrator
Narrowcast Administrator user role. This role is designed for a Narrowcast Administrator console user who performs some of the same tasks as the application designer but has the ability to modify application objects created by any Narrowcast Administrator account. This role is intended for quality managers, development managers, configuration managers, and other users requiring full access to all application objects. See also: application designer.
**application designer**
A Narrowcast Administrator user role. This role is designed for a Narrowcast Administrator console user who develops and tests services, which include documents, publications, schedules, and subscription sets. Configures and manages Subscription Portals that allow end users to subscribe to a variety of Narrowcast services via the Web. Configures data sources, content, and portal layout. Publishes services and device types, and selects default devices for Subscription Portals. Specifies information source properties and default site preferences. Installs, configures, and administers the development environment. Administers subscribers and subscriptions for development and testing purposes. See also: application administrator.

**application object**
MicroStrategy object used to provide analysis of and insight into relevant data. Application objects are developed in MicroStrategy Developer and they are the building blocks for reports and documents. Application objects include these object types: report, document, template, filter, metric, custom group, consolidation, prompt.

**application program interface (API)**
A set of related functions that provides an interface between existing applications and new applications. The API can be seen as a platform over a set of services on which new applications can be built. The functions, or interfaces, are implemented in a Dynamic Link Library and are defined in a standardized syntax. Application functionality available in the platform can be integrated or embedded into other applications through the use of the APIs.

**attachment**
Any file that is included in an email message. Attachments can originate from outside Narrowcast Server (that is, any pre-existing file can be sent as part of a service) or can be created by Narrowcast Server (that is, Narrowcast Server can create an Excel attachment on the fly).

**attribute**
A data level defined by the system architect and associated with one or more columns in a data warehouse lookup table. Attributes include data classifications like Region, Order, Customer, Age, Item, City, and Year. They provide a means for aggregating and filtering at a given level.

**attribute element**
A value of any of the attribute forms of an attribute. For example, New York and Dallas are elements of the attribute City; January, February, and March are elements of the attribute Month.

**attribute form**
One of several columns associated with an attribute that are different aspects of the same thing. Every attribute supports its own collection of forms.
**attrition rate**
The number of lost employees divided by the number of employees in a given time period.

**authentication object**
Object used by an information source that specifies who the user is and the security context within which that user will interact with that information source. Contains the security information required to make a connection or perform task execution. Depending on the information source module, it should contain information such as the user login name and password. For a MicroStrategy Information Source, the object is a MicroStrategy user. See also: personalization object.

**auto text code**
Dynamic text that is populated by the document or dataset, consisting of the document’s or dataset’s settings rather than data from the data warehouse. Examples of auto text codes, which can be considered as a type of variable, are document name, page number, and execution time. Auto text codes are contained in text field controls on a document. See also: Data field, Text field.

**autostyle**
A set of predefined formatting that can be easily applied to many reports in either MicroStrategy Developer or MicroStrategy Web. Autostyles are a good way to apply a corporate look and feel to reports.

**axes (axis)**
(1) A vector along which data is displayed. There are three axes—Row, Column, and Page. When a user defines a template for a report, he places template units—attributes, dimensions, metrics, consolidations, and custom groups—along each axis. (2) One part of a multi-part graphical diagram. Many SDAM reports display data on more than one graphical axis, such as the Quotation Activity Summary report and the Quarterly Conversion Summary report.

**B**

**banding**
A method of organizing values according to a set of descriptive or meaningful data ranges called buckets. Banding is also used for display purposes, where every other row is a different color and the two colors alternate. Compare: consolidation.

**base table**
A fact table that stores data at the lowest level of dimensionality.

**block**
A logical display element used to control the display of large reports in the limited page and slide dimensions of Microsoft Word and PowerPoint. A block may consist of multiple fetches. Blocks are defined by Microsoft Office product-specific configuration settings.
**break by**

An attribute or hierarchy where calculations for an OLAP function restart. To break by an attribute or hierarchy means to restart calculations that use OLAP, or Relative, functions when the analytical engine reaches the next instance of the specified attribute or hierarchy. Examples of OLAP functions include RunningStdevP, Rank, NTile, and various expressions that calculate percent values. To break by an attribute or hierarchy in an expression, you must set the BreakBy parameter.

**business intelligence (BI) system**

A system that facilitates the analysis of volumes of complex data by providing the ability to view data from multiple perspectives.

**cache**

A special data store holding recently accessed information for quick future access. This is normally done for frequently requested reports, whose execution is faster because they need not run against the database. Results from the data warehouse are stored separately and can be used by new job requests that require the same data. In the MicroStrategy environment, when a user runs a report for the first time, the job is submitted to the database for processing. However, if the results of that report are cached, the results can be returned immediately without having to wait for the database to process the job the next time the report is run.

**caching**

A special data storage method in which recently accessed values are stored for quick future access. Caching is used primarily to improve report execution performance.

**calculated expression**

A metric obtained dynamically, directly from metrics on a document dataset, by using at least one of the metrics in the document. Calculated expressions allow you to use simple arithmetic operators (+, -, *, /) to combine metrics from different datasets in the document. See also: Derived metric.

**catalog**

A table that contains the names of all non-temporary tables in a data warehouse.

**characteristic attribute**

An attribute that is a parent of a child attribute, but not part of the "main" hierarchy associated with the child attribute. For example, consider a hierarchy consisting of Year, Month, Day. Day of Week is a parent of Day, and a characteristic attribute.

**child attribute**

The lower-level attribute in an attribute relationship. See also: parent attribute or relationship.
clustering
A way of using machine resources to provide an efficient and robust processing environment for a Narrowcast Server system. A cluster consists of two or more machines, each machine running at least one Narrowcast Server component. These components are: MicroStrategy Logging Server, distribution manager (DM), execution engine (EE).

column
(1) A one-dimensional vertical array of values in a table. (2) The set of fields of a given name and datatype in all rows of a given table. (3) MicroStrategy object in the schema layer that can represent one or more physical table columns or no columns.

component
A computing unit that provides a specific piece of the Narrowcast Server functionality and interacts with other components. Examples are the Narrowcast Administrator, execution engine, distribution manager, MicroStrategy Logging Server, Object Repository, and Subscription Book Repository.

compound metric
A metric that cannot have a level placed on the entire metric, although it can be set separately on each of the components.

conditional formatting
A method used to format specified controls in a document depending on predefined criteria. It allows certain properties of controls, including sections, to be controlled by data-driven conditions.

conditional metric
A metric containing filter criteria in its definition.

conditionality
Conditionality of a metric enables you to associate an existing filter object with the metric so that only data that meets the filter conditions is included in the calculation.

configuration objects
A MicroStrategy object appearing in the system layer and usable across multiple projects. Configuration objects include (among others) these object types: users, database instances, database logins, schedules.

connection string
Stores the information required to connect to a database server. A connection string usually includes a DSN and the user ID and password required to log in to the database server. This information varies depending on the particular database server.

console user
A user who works with the Narrowcast Administrator console, managing subscriptions, developing services, and administering the system, in contrast to
an end user. See also: end user.

**consolidation**
An object that can be placed on a template and is made up of an ordered collection of elements called consolidation elements. Each element is a grouping of attribute elements that accommodates inter-row arithmetic operations. Compare: custom group.

**content**
The information in services, including all reports, miscellaneous text, and file attachments that are accessible to the user. Content is dynamic in the sense that personalization, error handling conditions, and device settings all influence the overall content output and format of each service.

**control**
Any item in the document’s Layout area that you can select. This can be a text field, line, rectangle, image, panel stack, selector, Grid/Graph, or HTML container. These different kinds of controls are referred to as control types. See also: Grid/Graph, HTML container, Panel stack, Selector, Text field.

**control default**
A set of options that can be set for each type of control and each section in a document. You can set the defaults according to the control that is currently selected; afterward, its format is applied to any object of the same type that you create in the document.

**custom group**
An object that can be placed on a template and is made up of an ordered collection of elements called custom group elements. Each element contains its own set of filtering qualifications.

**custom SQL**
Additional SQL code independently created by the user for execution against the data warehouse. MicroStrategy provides tools to write custom SQL, including Freeform SQL and Query Builder.

**Dashboard**
An interactive, visually intuitive display of data. A dashboard can summarize key business indicators (KPIs) to provide a status check. Users can change how they view the dashboard's data using interactive features, such as selectors, grouping, widgets, and visualizations. Users can explore their data via multiple paths, using text, data filtering, and layers of organization. See also: Visual Insight dashboard, Dashboard-style document.

**dashboard-style document**
A visually intuitive display of data that summarizes key business indicators for a quick status check. A special type of document, dashboard-style documents...
usually provide interactive features that let users change how they view the dashboard-style document’s data.

**Data Explorer**
A portion of the interface used to browse through data contained in the warehouse. Users can navigate through hierarchies of attributes that are defined by the administrator to find the data they need.

**data field**
Dynamic text that is populated from a dataset with data that originated in the data warehouse (or an Intelligence Server cache). A data field is only a reference to the metric, attribute, consolidation, or custom group on a report. Data fields are contained in text field controls on a document. See also: Auto text code, Text field.

**data mart**
A database, usually smaller than a data warehouse, designed to help managers make strategic decisions about their business by focusing on a specific subject or department.

**data mart report**
A special kind of report that saves its report data in a database rather than returning those results to the user. Data mart reports either create a new table in the database to store the report data or append the report data into an existing table.

**data modeling**
A method used to define and analyze data requirements needed to support the business functions of an enterprise. These data requirements are recorded as a conceptual data model with associated data definitions. Data modeling defines the relationships between data elements and data structures.

**data source**
A data source is any file, system, or storage location which stores data that is to be used in MicroStrategy for query, reporting, and analysis. A data warehouse can be thought of as one type of data source, which refers more specifically to using a database as your data source. Other data sources include text files, Excel files, and MDX cube sources such as SAP BW, Microsoft Analysis Services, Oracle Essbase, and IBM Cognos TM1.

**data source name (DSN)**
Provides connectivity to a database through an ODBC driver. A DSN generally contains host machine name or IP address, instance name, database name, directory, database driver, User ID, password, and other information. The exact information included in the DSN varies by DBMS. Once you create a DSN for a particular database, you can use it in an application to call information from the database.
**data warehouse**
A database, typically very large, containing the historical data of an enterprise. Used for decision support or business intelligence, it organizes data and allows coordinated updates and loads.

**database connection**
Stores all database-specific connection information such as DSN, driver mode and SQL execution mode as well as connection caching information.

**database instance**
(1) Database server software running on a particular machine. Though it is technically possible to have more than one instance running on a machine, there is usually only one instance per machine. (2) The MicroStrategy object that represents a logical definition of a data warehouse. It stores all information necessary for MicroStrategy to access the data warehouse for a particular project.

**database login**
The login ID and password that MicroStrategy Intelligence Server uses to log in to a particular database on behalf of a user. See also: login ID.

**dataset**
A MicroStrategy report used to define the data available on a Report Services document.

**Datasets**
(1) A pane in the Document Editor that shows all objects (grouped by datasets) that can be used in the document as supplied by the datasets. (2) All objects that can be used in the document as supplied by the datasets. Dataset objects are attributes, consolidations, custom groups, and metrics.

**decile (deciling)**
The method by which a group is broken up into ten groups of equal elements. The first decile consists of the top ten percent; the second, the 11th to 20th percent; the third, the 21st to 30th percent; and so on. demographics, demographic data. Data that locates, identifies, and describes a population and its properties; for example, data describing the age groups of people living in certain geographical areas or income categories. Other dimensions of demographic data include race, religion, political preference, spending preferences, and family size.

**demographics (demographic data)**
Data that locates, identifies, and describes a population and its properties; for example, data describing the age groups of people living in certain geographical areas or income categories. Other dimensions of demographic data include race, religion, political preference, spending preferences, and family size.

**derived attribute**
An attribute calculated from a mathematical operation on columns in a warehouse table. For example, Age can be calculated from the expression [Current Date–Birth Date]. See also: attribute.
**derived metric**  
A metric based on data already available in a report. It is calculated by the Intelligence Server, not in the database. Use a derived metric to perform calculations on report data after it has been returned from the database.

**description column**  
Optional columns that contain text descriptions of attribute elements.

**device**  
MicroStrategy object that represents message-receiving technology employed by end users, such as a mobile phone or tablet. The device object specifies how a publication should be formatted and transmitted to a specific device type. For example, an Outlook 98 Device might specify that the first document in the publication must be plain text and no longer than 128 characters. It might also specify that the second document could be either plain text or HTML, and if it is HTML, image references can be embedded. A device includes specifications for both formatting and transmission. Devices are used in the definition of addresses to specify what information transmitter will be used to transmit content to those addresses and how that content should be formatted and packaged. See also: address, content, information transmitter, administration object.

**device ID**  
Numeric ID value that indicates the delivery method and device that a recipient prefers. Devices are specified for subscriptions in dynamic subscription sets by providing the device ID for each recipient. Device IDs are found under the properties for each device within Narrowcast Administrator. For example, one subscription might specify Outlook 2000, while another recipient might specify a mobile device. One subscriber then only receives email content formatted for Outlook while the other subscriber only receives mobile device content formatted for this phone.

**dimension**  
An element or factor making up a complete entity or variable (a quantity that may assume any one of a set of values).

**directory server**  
A directory service provider running on a particular machine. Directory servers are often part of email servers, and stores user names, addresses, and authentication information. Unlike the Subscription Book, however, they are neither intended nor well-suited to store and retrieve subscriptions. Directory servers usually enable client connections through the lightweight directory access protocol (LDAP), and are often used for centralized user authentication across many systems. See also: Subscription Book, Lightweight Directory Access Protocol (LDAP).

**display locale**  
Controls which object names are shown in Narrowcast Administrator. Since objects can have multiple names to support end users in more than one language, one name must be chosen for display in Narrowcast Administrator. The display locale serves this role. The locale selected for this purpose should be
the one that the Narrowcast Administrator needs to use most frequently when defining objects. See also: locale, execution locale, selection locale, system locale, user locale.

distribution manager (DM)
(1) Delivery Engine object that receives service execution triggers, distributes service data to the execution engines for processing, designates failover execution engines, and tracks the status of other system components. If the primary distribution manager fails, one of the backup distribution managers becomes the primary and takes over processing where the failed component stopped. (2) Piece of software or component used to instantiate a distribution manager object. (3) Machine being used to instantiate the distribution manager object.

distribution set
A schedule and subscription set pair, created during service definition. Specifies when and to whom Narrowcast Server will send a service. See also: schedule, subscription set.

document
A display representing data coming from one or more reports, as well as positioning and formatting information. A Report Services document is used to format data from multiple reports in a single display of presentation quality.

document (Narrowcast)
A Narrowcast document contains the static information from the document template and the dynamic content from document elements, and is ready to be transmitted by Narrowcast Server. Each document has a specific type, such as HTML, plain text, Excel, or derived from an imported file.

document (Report Services)
A type of data display that shows data usually coming from multiple reports. A Report Services document formats data from multiple reports in a single display of presentation quality. A dashboard is a type of Report Services document, often including interactive components.

document element
Part of a Narrowcast document containing dynamic content that is generated when that document is executed as part of a service. Consists of at least one information object, plus error-handling instructions. Might include an XSL stylesheet selection, depending on the type of information object chosen.

document formatter
Transforms the raw data of the information objects and the structure of the Narrowcast document templates into a complete, formatted Narrowcast document. The operations of document formatters include such tasks as XSL processing and transformation, XML merging, character replacement, and string padding. Formatting rules might account for device characteristics, locale definition, and other recipient-specific formatting control information.
document formatting module (DFM)
A piece of software, or a component, that performs the work required for turning Narrowcast document templates and information objects into Narrowcast documents. The formatting process can use device characteristics, locale definition, and other recipient-specific control information.

document template
Provides the structure and layout for a Narrowcast document. For example, an Excel document template includes the basic workbook structure, any predefined macros, and static worksheet content.
drill
A method of obtaining supplementary information after a report has been executed. The new data is retrieved by re-querying the Intelligent Cube or database at a different attribute or fact level.
drill path (attribute drill path)
In MicroStrategy, a path that determines which attributes are presented to an interface; typically a project defines drill paths from parent attributes to their children.
dynamic content
Document content that is dynamically retrieved at service execution time and that can be personalized for each message recipient. Dynamic content is created using content information objects. This content changes depending on the results returned by the information object and can return different results for different subscribers. For example, a weekly report returns different information from week to the next, and each subscriber might request different content in his report. Examples include a MicroStrategy report, a Web query, a SQL query, or content from an external system. See also: static content.
dynamic subscription set
A subscription object containing at least one piece of subscription information retrieved from an information object instead of from the Subscription Book Repository. Useful for changing or alert-driven subscription sets. Some dynamic subscription sets acquire all their information from a single information object. Others retrieve only the Subscription ID from the information object while the rest of the subscription information is acquired from the Subscription Book Repository. Still others combine both methods. See also: segmentation information object, Subscription ID, subscription information object, subscription set.
editor
A dialog used to create and edit MicroStrategy Objects. There is a Filter Editor, Template Editor, Attribute Editor, Metric Editor, Report Editor, and so on.
end user
A subscriber, the person receiving messages from Narrowcast Server service, in contrast to a console user. See also: console user.

entity relationship diagram (ERD)
A diagram that provides a graphical representation of the physical structure of the data in the source system, which lets you easily recognize tables and columns and the data stored in those columns.

entry level
The lowest level set of attributes at which a fact is available for analysis.

ETL
Short for extraction, transformation, and loading. (1) The process used to populate a data warehouse from disparate existing database systems. (2) Third-party software used to facilitate such a process.

execution engine (EE)
(1) Delivery Engine object that controls the execution of narrowcast messages, called services. It receives service segments from the distribution manager; then determines the content and subscriber information with input from other components, including the Subscription Book Module and information source modules. (2) A piece of software or component that is used to instantiate an execution engine object. (3) A machine being used to instantiate an execution engine object.

execution locale
Determines how content is generated for the users who receive it. When a document is created, the application designer can specify that the document should be executed in a specific execution locale. The definition of the locale then controls how information should be gathered, formatted, packaged and delivered for this document. For example, an application designer might define a French and a German locale to gather information in the appropriate languages. If one of these locales is chosen as the execution locale for a document, all dynamic content in this document is retrieved using the corresponding language. See also: locale, display locale, selection locale, system locale, user locale.

expression
Formulas built from functions, attributes, facts, metrics, and consolidations that can be used to define attribute forms, fact calculations, metrics, or filters.

fact
(1) A measurement value, often numeric and typically aggregatable, stored in a data warehouse. (2) A schema object representing a column in a data warehouse table and containing basic or aggregated numbers—usually prices, or sales in dollars, or inventory quantities in counts. See also: metric.
**fact table**
A database table containing numeric data that may be aggregated along one or more dimensions. Fact tables may contain atomic or summarized data. Compare: base table.

**fetch**
The amount of report data retrieved from one call to MicroStrategy Web Services. Fetches are used to control the amount of network traffic between MicroStrategy Office and MicroStrategy Web Services and the amount of memory used by MicroStrategy Intelligence Server. When you execute a report or document into a Microsoft product, the Execution Status dialog box displays the progress of each fetch. Fetch size is controlled by Microsoft Office product-specific configuration settings. Maximum fetch size is governed by an Intelligence Server setting in MicroStrategy Developer.

**filter**
A MicroStrategy object that specifies the conditions that the data must meet to be included in the report results. Using a filter on a report narrows the data to consider only the information that is relevant to answer your business question, since a report queries the database against all the data stored in the data warehouse. A filter is composed of at least one qualification, which is the actual condition that must be met for the data to be included on a report. Multiple qualifications in a single filter are combined using logical operators. Examples include "Region = Northeast" or "Revenue > $1 million". A filter is normally implemented in the SQL WHERE clause.

**Flash-enabled document**
A Report Services document in which Flash Mode is selected as an available display mode in the Document Properties dialog box.

**flattened**
A report display type in which the results of a report are displayed with all attributes and metrics flattened side-by-side on one axis. Also, any subtotals within the report are not displayed. It is convenient to execute your report or HTML document in this display type if you intend to use Excel’s drop-down lists to filter, hide/show data, and more.

**folder**
A MicroStrategy object used for grouping and storing in a single place a set of objects that are similar, such as filters, templates, and reports.

**folder list**
A portion of the interface that lists all the folders found in the project in a hierarchical fashion. It helps a user to visualize and browse through a MicroStrategy project.

**form**
One of several columns that are different representations of the same thing, as ID, Name, Long Description, Abbreviation.
function plug-in
Allows user-defined functions to be used by the MicroStrategy Analytical Engine.

governing parameters
Designed to keep the execution profile of a system within acceptable limits. Protects the system from overconsumption of resources such as memory, disk space, network capacity, and CPU cycles.

graph
A graphical image display of data. Sometimes referred to as a chart. See also report.

graph analytic
An analytic showing data as points, lines, or bars, arranged according to axes based on the chosen metrics. Although not all analytics can be displayed in every graph type available, many analytics can be viewed in several ways. Choices for graph display include bars, lines, area graphs, and three-dimensional graphs.

graph view
Report viewing mode that displays results as a graphical chart.

grid
A spreadsheet-style two-dimensional display of data. See also report.

grid analytic
An analytic consisting of rows and columns. Each row or column has a heading, and each heading represents a prompt from the analytic. Grid analytics can be modified easily, by drilling, moving columns, converting columns to rows, sorting, and using page-by to display subsets of the analytic data as separate pages.

grid view
Report viewing mode that displays result data in a tabular format.

Grid/Graph
A control placed in a document that displays information in the same way a MicroStrategy report does.

grouping
A way to create a hierarchical structure for a document.

hierarchy
A set of attributes defining a meaningful path for element browsing or drilling. The order of the attributes is typically—though not always—defined such that a higher attribute has a one-to-many relationship with its child attributes.
**Hint**
A comment that passes instructions to a database optimizer about choosing an execution plan for a given SQL statement. In MicroStrategy, a hint can be defined in VLDB properties to appear within a MicroStrategy-issued SQL statement.

**History List**
A folder where users can retrieve the results of previously executed or scheduled reports and documents.

**HTML container**
A control that either displays real-time information from the web or displays formatted HTML.

**HTML document**
1. A compound report displaying multiple grids and graphs. 2. The MicroStrategy object that supports such a report.

**ID column**
A column that contains attribute element identification codes. All attributes must have an ID column.

**Imported file**
A file imported from outside of Narrowcast Server and stored in its original format. For example, a text file contains only plain text and an HTML file contains only HTML. Storing information in the file format preserves the integrity of the original data since files cannot be modified with Narrowcast Server. Files cannot be directly included in publications; they must first be associated with imported documents, which serve as containers to allow them to be attached to publications. Imported files can also be used as stylesheets or templates for documents.

**Incremental fetch**
A feature that returns a large set of information, such as subscriptions, to the console in numerous small pieces as those pieces are needed. This ensures that network traffic and client processing is kept to a minimum.

**Information delivery platform**
Server that, whenever certain conditions are in effect, acquires information from a variety of sources, personalizes that information, formats it, and transmits it through a variety of technologies.

**Information object**
Application object containing a set of instructions that specify how to get data from a particular type of information source. Defined by using an information source module, it is set up when a service is built and executed when the service that uses it is run. Plays one of three roles: content, subscription, or
segmentation. See also: information object role, information object type, information source, information source module (ISM).

**information object role**
Indicates how the information supplied by the information object is used by Narrowcast Server. Three roles are available: Content (Supplies information for the document), Subscription (Provides a list of subscribers and related subscription information for dynamic subscription sets), Segment (Specifies the first and last subscriptions for each segment of the subscription set). See also: information object.

**information object type**
Indicates the format of the data returned by the information object and how Narrowcast Server can use this data. Three types are available: Text (Returns a plaintext string that is inserted directly into a document's content), XML (Returns an XML string that must be combined with an XSL stylesheet to produce content appropriate for each type of document), Image (Returns an image in JPG format that is inserted directly into the document's content). See also: information object.

**information source**
Contains information about how to connect to a specific source of content using an information source module. The Delivery Engine requests and acquires information from information sources. An information source uses personalization objects for personalization. See also: information source module (ISM), personalization object.

**information source module (ISM)**
An executable process that receives requests for information and returns that information as well-defined data. It defines and processes information objects. One ISM can be used to load various information sources. For example, an ISM can be set up to access MicroStrategy Developer projects and can be used for any MicroStrategy Developer projects that you want to use as an information source. See also: information object, information source.

**information transmission module (ITM)**
An executable process or component designed to support a particular delivery capability. For example, MicroStrategy Narrowcast Server provides the email (SMTP) Module, which is used for the email (SMTP) Information Transmitter. The ITM indicates what documents and document types it supports, as well as the arrangement of these documents. Also, the ITM provides a user interface for defining device characteristics. Those characteristics are stored in the Object Repository and retrieved by the information transmission module to control packaging and delivery. See also: device, document (Narrowcast), information transmitter, publication.

**information transmitter**
Administration object that delivers formatted content, in the form of publications, to end user devices. An information transmitter might serve the purpose of sending email via SMTP or mobile device messages using an SMS gateway. In general, an information transmitter supports one type of
transmission technology, such as SMTP, FTP, HTTP, or ODBC, but this is not a requirement. Information transmitters depend on information transmission modules. See also: device, document (Narrowcast), information transmission module (ITM), publication.

**Installation log file**
The MicroStrategy 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.

**Intelligent Cube**
A copy of the report data saved in memory and used for manipulation of the view definition. This division allows multiple reports with different views to share a common data definition.

**Java database connectivity**
A Java API that enables Java programs to execute SQL statements. This allows Java programs to interact with any SQL-compliant database. Since nearly all relational database management systems (DBMSs) support SQL, and because Java itself runs on most platforms, JDBC makes it possible to write a single database application that can run on different platforms and interact with different DBMSs. JDBC is similar to ODBC, but is designed specifically for Java programs, whereas ODBC is language-independent.

**join**
A SQL operation that combines data from multiple tables into a single result table.

**KPI (key performance indicator)**
An indicator gauging how well a company progresses in numerous areas such as finance, customer service, and product availability and distribution.

**Layout area**
The middle panel of the Document Editor in which you place data or other controls to determine the appearance of the document when it is viewed as a PDF.

**level**
(1) In a data warehouse, facts are said to be stored at a particular level defined by the attribute IDs present in the fact table. For example, if a fact table has a Date column, an Item_ID column, and a fact column, that fact is stored at the Date/Item level. (2) With regard to metric calculation, the level is the level of calculation for the metric.
**Lightweight Directory Access Protocol (LDAP)**
An open standard that client computers use to communicate with directory service providers. Client machines connect to a particular logical directory on a particular physical directory server. See also: directory server.

**Lightweight Directory Interchange Format (LDIF)**
File format that is exported from LDAP directory services.

**link**
A connection from a document to another document or a report. A link lets an analyst execute another document or report (the target) from a document (the source), and to pass parameters to answer any prompts that are in the target.

**load balancing**

**locale**
Specifies what regional characteristics to apply to data and formatting, including number format, date format, time format, and character sets. Your system might support multiple locales. See also: display locale, execution locale, selection locale, system locale, user locale.

**Locale ID**
Numeric ID value used to indicate the locale that a recipient prefers. Locales are specified for subscriptions in dynamic subscription sets by providing the Locale ID. Locale IDs are found under the properties for each locale within Narrowcast Administrator. For example, one subscription might specify French, while another recipient might specify German. One subscriber then only receives French content while the other subscriber only receives German content.

**logical data model**
A graphical representation of data that is arranged logically for the general user, as opposed to the physical data model or warehouse schema, which arranges data for efficient database use.

**login ID**
Login identifier, typically supplied with a password.

**lookup table**
A database table used to uniquely identify attribute elements. They typically consist of descriptions of dimensions. Lookup tables are usually joined to fact tables to group the numeric facts in the fact table by dimensional attributes in the lookup tables.
managed object
A schema object unrelated to the project schema, which is created by the system and stored in a separate system folder. Managed objects are used to map data to attributes, metrics, hierarchies and other schema objects for Freeform SQL, Query Builder, and MDX cube reports.

manual fetch
A type of incremental report result fetching in which you are asked to confirm each fetch. For example, when you execute a report, you are prompted to confirm that each fetch, or section of report results, is displayed.

many-to-many relationship
An attribute relationship in which multiple elements of a parent attribute can relate to multiple elements of a child attribute, and vice versa.

many-to-one relationship
An attribute relationship in which (1) multiple elements of a parent attribute relate to only one element of a child attribute, and (2) every element of the child attribute can relate to multiple elements of the parent.

MDX cube
An MDX cube is a collection or set of data retrieved from an MDX cube source, which is imported into MicroStrategy and mapped to various objects to allow query, reporting, and analysis on the data.

MDX cube report
The central focus for MicroStrategy users to query, analyze, and visually present data from MDX cube sources in a manner that answers and evaluates their business questions. MDX cube reports provide the same data display and analysis functionality as standard MicroStrategy reports, but rather than reporting on data from a relational data warehouse, MDX cube reports report on data from MDX cube sources.

MDX cube source
When integrated with MicroStrategy, the third-party tools SAP BW, Microsoft Analysis Services, Oracle Essbase, and IBM Cognos TM1 are referred to as MDX cube sources. You can import and map data from these different MDX cube sources in MicroStrategy to query, report on, and analyze data with MicroStrategy. MicroStrategy can integrate with MDX cube source data as well as access data from a relational database concurrently.

messaging application program interface (MAPI)
Allows Windows applications to send email messages through external email programs. Designed primarily to connect client applications such as Microsoft Outlook to mail servers such as Microsoft Exchange Server. Not intended for server-to-server communications.
metadata
A repository whose data associates the tables and columns of a data warehouse with user-defined attributes and facts to enable the mapping of the business view, terms, and needs to the underlying database structure. Metadata can reside on the same server as the data warehouse or on a different database server. It can even be held in a different RDBMS.

metadata (or metadata repository)
A repository whose data associates the tables and columns of a data warehouse with user-defined attributes and facts to enable the mapping of the business view, terms, and needs to the underlying database structure. Metadata can reside on the same server as the data warehouse or on a different database server. It can even be held in a different RDBMS.

metric
(1) A business calculation defined by an expression built with functions, facts, attributes, or other metrics. For example: Sum(dollar_sales) or [Sales] - [Cost].
(2) The MicroStrategy object that contains the metric definition. It represents a business measure or key performance indicator. See also: fact.

MicroStrategy Analytics Module
A MicroStrategy project with prepackaged metadata, including best practices reports, scorecards, and dashboards, key performance indicators, attributes, business metrics, filters, and custom groups; default physical and logical data models to allow the module to work with your physical schemas and data model or with the module’s packaged data warehouse schema; and a reference guide for the Analysis Module’s data model, the analysis area, metadata object definitions, data dictionary, and individual report use scenarios.

MicroStrategy Intelligence Server
Core of the MicroStrategy architecture, MicroStrategy Intelligence Server manages and organizes users, projects, and database connections; coordinates, prioritizes, and executes all user requests; and allocates the resources necessary to complete them. It tracks schedules, manages security, and provides the ability to monitor and analyze the daily activity of the entire decision support environment.

MicroStrategy Logging Client
Service that receives logging messages from Narrowcast Server components and relays them to the MicroStrategy Logging Server. Logging messages include segment status information, statistics information, and error event notification. See also: MicroStrategy Logging Server.

MicroStrategy Logging Server
Server that collects logging messages from MicroStrategy Logging Clients and distributes them to consumers. Logging messages include segment status information, statistics information, and error event notification. See also: MicroStrategy Logging Client.
multidimensional analysis
A form of analysis of the data in a data warehouse that includes many relationships, each representing a dimension. For example, a retail analysis may seek to understand the relationships among sales by region, by quarter, by demographic distribution (income, education level, gender), and by product. Multidimensional analysis provides results for these complex relationships.

multipart MIME
Part of the MIME specification, which includes more than one body part in a body section within a MIME message. This is typically used to enable the inclusion of separate text and HTML message bodies, embedded images, and other body parts within an email message. See also: multipurpose Internet mail extensions (MIME).

multipurpose Internet mail extensions (MIME)
The specification as defined by RFC 1521 (maintained by the Internet Engineering Task Force at http://www.ietf.org/) for encoding message contents, attached files, embedded images, and other embedded files into a single (typically 7-bit) data string. Many modern email clients support MIME. Because the MIME specification includes encoding of message contents, it allows for messages that use character sets other than US-ASCII. See also: multipart MIME.

N

Narrowcast Administrator account
An object that allows a user to log in to MicroStrategy Narrowcast Administrator. It is created within Narrowcast Administrator, and its definition is stored in the Object Repository of a system. Each Narrowcast Administrator account is based on a specific Windows account. When a new Narrowcast Server system is created, two Narrowcast Administrator accounts are automatically added to it, one of which is the local Administrator Windows account for the machine that was used to create the system. Narrowcast Administrator accounts can have various privileges, which control what Narrowcast Administrator users can do. These privileges include: Service design, Application administration, System administration, Subscription administration.

Narrowcast Server system
A collection of machines, software components, and objects that collectively provide the ability to process and deliver narrowcast messages to recipients. The objects that make up a system are divided into two categories, administration objects and application objects. See also: administration object, application object, clustering, component.

O

object
Conceptually, an object is the highest grouping level of information about one concept, used by the user to achieve the goal of specified data analysis. More
concretely, an object is any item that can be selected and manipulated, including folders, reports, facts, metrics, and so on.

**object template**
A MicroStrategy object that allows you to start with a predefined structure when creating a new object. You can use object templates for many MicroStrategy objects, including metrics, documents, reports, and report templates.

**ODBC (open database connectivity)**
An open standard with which client computers can communicate with relational database servers. Client machines make a connection to a particular logical database, on a particular physical database server, using a particular ODBC driver.

**ODBC driver**
Software interface to an ODBC service provider. See also: open database connectivity (ODBC).

**ODBC driver manager**
Coordinates communication between a client application and database server. The client application tells the driver manager that it needs to connect using a particular connection string. The DSN found in this connection string provides the driver manager with the type of database server to which the application needs access. From this information, the driver manager decides what driver to use and initiates the communication.

**one-to-many relationship**
An attribute relationship in which every element of a parent attribute can relate to multiple elements of a child attribute, while every element of the child attribute relates to only one element of the parent. The one-to-many attribute relationship is the most common in data models.

**one-to-one relationship**
An attribute relationship in which every element of the parent attribute relates to exactly one element of the child attribute, and vice versa.

**online analytical processing**
In general, a system with analytical processing that involves activities such as manipulating transaction records to calculate sales trends, growth patterns, percent to total contributions, trend reporting, and profit analysis.

**operational data store**
A database that typically stores transactional data generated by and used in the conducting of business operations. The data can be used as a trigger condition or as content.

**outline mode**
Report viewing mode that creates indented, collapsible groupings of related elements to make reports neater and easier to read.
page
An amount of data that is analogous to a page in a page-by report. Each page of a page-by report represents one combination of elements from each attribute on the page-by axis. For example, one page may contain data related to 2006 as well as the Southeast region. A report with no attributes on the page-by axis is considered to have one page. When you execute a report or document into a Microsoft product, the Execution Status dialog box displays the progress of each page.

page-by
Segmenting data in a grid report by placing available attributes, consolidations, and metrics on a third axis called the Page axis. Since a grid is two-dimensional, only a slice of the cube can be seen at any one time. The slice is characterized by the choice of elements on the Page axis. By varying the selection of elements, the user can page through the cube.

panel
A way of grouping data in a document so that users can navigate subsets of data as if the subsets were pages in a smaller document. Each “page”, or layer of data, is a panel; a group of panels is called a panel stack.

panel stack
The holder for a collection of panels, or layers of data, in a document. A user can navigate or flip through the panels in a panel stack; only one panel is displayed at a time.

parent attribute
The higher-level attribute in an attribute relationship with one or more children. See also: child attribute or relationship.

partition
A relational database table broken down into smaller component tables. This can be done at the database level or at the application level. See the MicroStrategy System Administration Guide for more information.

partition mapping
The division of large logical tables into smaller physical tables based on a definable data level, such as month or department. Partitions minimize the number of tables and records within a table that must be read to satisfy queries issued against the warehouse. By distributing usage across multiple tables, partitions improve the speed and efficiency of database queries.

partition mapping table
A warehouse table that contains information used to identify the partitioned base tables as part of a logical whole. (A partitioned base table is a warehouse table that contains one part of a larger set of data. Partition tables are usually divided along logical lines, such as time or geography.) Also referred to as a PMT.
persistence
A default behavior in MicroStrategy Office ensuring that every report and document you run in an Excel workbook, PowerPoint presentation, or Word document retains a link to MicroStrategy Intelligence Server. This ensures that you and other users can refresh the reports and documents to retrieve the latest data from the data warehouse or other data sources. Every report or document executed also contains properties that are persisted; these properties determine how the report or document is displayed and can be modified by users once it is refreshed.

personalization
The process that allows each subscriber to receive data that has been filtered and formatted specifically for him. Narrowcast Server uses locale, subscription, and personalization objects to personalize services. See also: locale, personalization object, subscription.

personalization object
An object handled by information sources for personalization. The three types are: authentication, preference and security. The combination of the authentication, preference, and security objects control the data that is returned by an information object. See also: authentication object, information object, information source, preference object, question object, security object.

personalization set
A combination of locale, authentication, security, and preference objects to achieve personalized information, or information filtered and formatted specifically for a subscriber.

personalized page execution
In the context of the MicroStrategy Information Source Module, each page of a report is used as a result for one or more subscribers. See also: page-by, personalization set.

personalized report execution
In the context of the Information Source Module, each result is derived from the separate execution of a report. Answers to prompts and MicroStrategy users both impact personalization in this execution mode. See also: personalization set.

physical address
The address location used to locate recipients and deliver messages. For email delivery, this is a standard SMTP address in the form recipient@domain.xxx. No physical address is required for delivery to the Subscription Portal. The physical address required by other information transmitters depends on the individual information transmitter.

physical warehouse schema
A detailed graphic representation of your business data as it is stored in the data warehouse. It organizes the logical data model in a method that make sense from a database perspective.
PIN
The personal identification number required by some delivery methods. A PIN is not used for email, mobile device, or Subscription Portal delivery. The use of a PIN by other information transmitters depends on the individual information transmitter.

plug-in
An additional processing component integrated with MicroStrategy products. For example, advanced statistical and financial functions can be added as additional processing components.

plugin
An application that can easily be installed and executed by the MicroStrategy Intelligence Server in one of several identical interfaces. For example, advanced statistical and financial functions can be added as additional processing components.

port number
The port number is how a server process identifies itself on the machine on which it is running. For example, when the Intelligence Server machine receives a network call from a client (Developer, Web Universal, Narrowcast Server, Command Manager, and so on), it knows to forward those calls to the Intelligence Server port number that is specified in the call.

portability
The ability of an analytical application to be integrated into an existing data warehouse. To port the Analysis Module, you “map” the module to the physical schema of an existing data warehouse.

portal
A site that offers a centralized access point for finding and managing information via a variety of different services. It offers a broad array of resources and services, such as email, discussion forums, search engines, and other online services. A portal is accessible through the use of a web browser. See also: Subscription Portal.

preference object
Type of personalization object that uses the answers to question objects to specify the information that the user wants. At run-time, the preference object is applied to information objects to personalize them. Preferences are usually controlled by the user, but can be set by the subscription administrator. For an Information Source, preference objects are prompt answers. See also: personalization object, question object.

prefix
A prefix is stored in the project metadata associated with a table or tables and is used by the Engine to generate SQL. Also, the Catalog Server uses it to obtain table sample values and row counts. In most cases, it should match the name space field since it is used to qualify on a specific table belonging to a certain
owner or name space. Prefixes can be defined and modified from the Warehouse Catalog interface.

**preview fetch**
A type of incremental report result fetching in which you can see a preview of the report data as it runs. You specify how many rows of the report you want to preview at a time, and fetching automatically stops after one fetch.

**primary key**
In a relational database, the set of columns required to uniquely identify a record in a table.

**privilege**
Assigned to users, a privilege defines the functionality available to a user, for example, which objects a given user can create and which applications and editors he can use.

**process**
An executing application comprising one or more threads. Processes use temporary private address spaces and control operating system resources such as files, dynamic memory allocations, pipes, and synchronization objects.

**production metadata**
The repository you create during the configuration portion of the installation process, and which works with your data warehouse and serves as your working metadata repository.

**productivity analysis**
A process that measures company productivity and identifies ways to increase it.

**project**
(1) The highest-level intersection of a data warehouse, metadata repository, and user community, containing reports, filters, metrics, and functions. (2) An object containing the definition of a project, as defined in [1]. The project object is specified when requesting the establishment of a session.

**project designer**
The user category of one who creates projects and all of the schema objects (facts, attributes, hierarchies) for a project. A project designer is thoroughly familiar with the data model and schema object editors.

**project source**
Defines a connection to the metadata database and is used by various MicroStrategy components to access projects. A direct project source is a two-tier connection directly to a metadata repository. A server project source is a three-tier connection to a MicroStrategy Intelligence Server. One project source can contain many projects and the administration tools found at the project source level are used to monitor and administer all projects in the project source.
prompt
MicroStrategy object in the report definition that is incomplete by design. The user is asked during the resolution phase of report execution to provide an answer that completes the information. A typical example with a filter is choosing a specific attribute on which to qualify.

Property List
The list of settings used to specify the appearance or any other characteristic of a control on a document.

publication
An ordered collection of documents that completely defines the content of a service for a specific set of devices. Each publication is used for exactly one locale and one information transmission module. A publication specifies: An information transmitter, At least one device or A set of documents. See also: device, document (Narrowcast), information transmitter.

Q

qualification
The actual condition that must be met for data to be included on a report. Examples include “Region = Northeast” or “Revenue &gt; $1 million”. Qualifications are used in filters and custom groups. You can create multiple qualifications for a single filter or custom group, and then set how to combine the qualifications using the logical operators AND, AND NOT, OR, and OR NOT.

query
A request for data from a database or data warehouse. A report is a database query.

Query Engine
The MicroStrategy component responsible for submitting SQL code to the database.

question object
Type of personalization object that specifies the questions to ask the user to determine the user’s preferences. Answers are used as preference objects. For a MicroStrategy Information Source, a question object is defined by choosing a report, and the questions to be asked depend on the definition of that report. In personalized page execution mode, the user is asked what page to display, using the page-by-functionality. In personalized report execution mode, the questions include all prompt objects defined on the selected report except the security object prompt. See also: page-by, personalization object, preference object.

Quick Grid
A report display type in MicroStrategy Office in which the report is run as a CSV and bulk-loaded into Excel in one fetch. Although this results in a fast report execution time, formatting from the report definition, such as fonts, colors, and thresholds is not applied. You can apply formatting to Quick Grids using Microsoft Excel’s AutoFormats.
**R**

**ranking**
A type of OLAP function that returns the rank of a value in a group of values. Rows with equal values with respect to the ordering are assigned the same rank.

**relate table**
A table containing the ID columns of two or more attributes, thus defining associations between them.

**relationship**
An association specifying the nature of the connection between one attribute (the parent) and one or more other attributes (the children). See also: child attribute or parent attribute.

**report**
The central focus of any decision support investigation, a report allows users to query for data, analyze that data, and then present it in a visually pleasing manner.

**report designer**
The user category of one who creates all application objects such as grid and graph reports, filters, templates, documents, consolidations, and custom groups. The report designer understands all of the business intelligence capabilities of the system.

**report resolution**
The process of filling placeholders such as prompts with information determined at run time.

**report template**
A MicroStrategy object that allows you to define the layout of general categories of information in a report. In a report template, you specify the information that you want to retrieve from your data source, and the way that you want the data to be displayed in Grid view. A report template does not include filter information. Report templates are often referred to as just as templates.

**retention(employee)**
The process of maintaining or securing employee loyalty to minimize loss of key talent.

**role**
A feature of the security subsystem that defines which objects a given user can create and which operations he can perform. The different Narrowcast Server roles are: System administrator (who manages the entire system), Application administrator (who administers application objects only), Application designer (who develops application objects only and cannot modify objects owned by other users), Subscription administrator (who manages the Subscription Book), Portal administrator (who configures and manages Subscription Portals).
schedule
Sets the time or frequency that a service is executed. A schedule represents a recurrence pattern, not a fixed date. It is defined relative to time zones to account for daylight savings, date boundaries, and other time zone-specific issues. To allow subscribers globally to receive services at specific local times, a service can contain more than one schedule.

schema
(1) The set of tables in a data warehouse associated with a logical data model. The attribute and fact columns in those tables are considered part of the schema itself. (2) The layout or structure of a database system. In relational databases, the schema defines the tables, the fields in each table, and the relationships between fields and tables.

schema object
MicroStrategy object created, usually by a project designer, that relates the information in the logical data model and physical warehouse schema to the MicroStrategy environment. These objects are developed in MicroStrategy Architect, which can be accessed from MicroStrategy Developer. Schema objects directly reflect the warehouse structure and include attributes, facts, functions, hierarchies, operators, partition mappings, tables, and transformations.

scorecard
A type of tally sheet displaying a company's performance using key performance indicators (KPIs) that gauge how well a company progresses in areas such as finance, customer service, and product availability and distribution. See also KPI (key performance indicator).

security filter
A qualification associated with a user that is applied to all queries executed by that user.

security object
Type of personalization object that specifies what information the user should have access to. Security filtering criteria applied to an information source during subscriber interaction. For a MicroStrategy Information Source, it is a response to a prompt in the form of a single attribute element. Each user can have one security object per information source. See also: personalization object, security object prompt.

security object prompt
Specifies which attribute in the project corresponds to a user or subscriber. The answer to this prompt is used as a security object in MicroStrategy. See also: prompt, security object.

security role
In a MicroStrategy security model, the set of privileges that a user can have.
**segment**
A group of subscriptions within a subscription set. Subscription sets are divided into multiple pieces, or segments, so that the work required to execute a service for all subscriptions can be distributed across multiple systems to allow parallel work processing. The size of the segment is part of the service definition and controls the work packages that are sent to each execution engine.

**segmentation**
The task of dividing the subscriptions within a subscription set into equal-sized groups or segments. Segmentation ensures that all subscription information objects and content information objects return only information for subscriptions in the current segment. Subscription sets are always segmented using the subscription ID, and the boundaries of segments are specified as subscription ID values. For the MicroStrategy Information Source, segmentation can be performed automatically by Narrowcast Server, or the application designer can control how the subscription set is segmented and how subscription information objects and content information objects constrain the information they return.

**segmentation information object**
An information object that returns the segment boundary values for a subscription set. Boundary values determine the size of the segment and are always given in the form of subscription ID values corresponding to the first subscription ID in each segment. Segment information objects are used only for dynamic subscription sets.

**segmentation prompt**
A prompt that controls the information returned in a report used as a content information object or subscription information object. Segmentation can be performed automatically by Narrowcast Server, or prompts can be added to content information objects and subscription information objects manually. This process limits the number of subscriptions and content pages returned to only provide information for subscriptions in the current segment.

**selection locale**
Determines which users should receive which content. When a user is created, the subscription administrator (using Narrowcast Administrator) or the user (using the Subscription Portal) selects a locale for which the user will receive content. When a publication is created, the application designer chooses a selection locale that determines which users receive the content in a publication. If a user chooses Locale A, the application designer must choose Locale A as the selection locale for a publication for the user to receive this publication. See also: locale, display locale, execution locale, system locale, user locale.

**selector**
A type of control in a document that allows a user to: Flip through the panels in a panel stack, to see different predefined layers of data, or “pages”, in the same document. Display different attribute elements or metrics in a Grid/Graph.
SequeLink
Third-party (non-MicroStrategy) software that configures and manages data access across multiple data stores, operating systems, and deployment options.

SequeLink machine
The machine where SequeLink is installed. This machine can be independent from the rest of the Subscription Portal.

SequeLink services
SequeLink creates two NT services: SLAgent 54, which is the Administrator, and SLSocket54, which is the Server.

server definition
A MicroStrategy object stored in the metadata containing information about the configuration of an Intelligence Server.

server instance
The combination of an Intelligence Server running with a particular server definition.

service
An object that provides all the information needed for the Delivery Engine to correctly generate messages. That information includes the delivery conditions, content, personalization rules, and subscriptions for sending messages for a particular purpose. For example, one service delivers a daily message containing a stock portfolio update, while another delivers alerts to mobile devices when the value of a stock changes significantly. A service comprises at least one schedule-subscription pair and a set of publications.

service designer
Narrowcast Administrator user role. This role is for a Narrowcast Administrator console user who develops and tests services, which include documents, publications, schedules, and subscription sets. Configures and manages Subscription Portals that allow end users to subscribe to a variety of Narrowcast services via the web. Configures data sources, content, and portal layout. Publishes services and device types, and selects default devices for Subscription Portals. Specifies information source properties and default site preferences. Installs, configures, and administers the development environment. Administers subscribers and subscriptions for development and testing purposes. See also: application administrator.

service provider interface (SPI)
The specification for a set of related functions that can be implemented by a software developer to augment or enhance the capabilities of a software product or platform. This allows a developer to enable his software to be called by an existing piece of software. See also: application program interface (API), plug-in.

service queue
In Narrowcast Administrator, a visual display of upcoming services to be processed.
**simple metric**
A type of metric that can stand alone or be used as a building block for compound metrics. Simple metrics always contain at least one aggregate function, such as sum or average, applied to a fact, attribute, or another metric. The entire metric can only contain one level.

**slice**
One page of content in a multi-page report. Narrowcast Server divides a single multi-page report into multiple individual pages of content that are used as personalized content for individual subscriptions. Slicing is an attribute. A slicing attribute is the attribute used to divide multi-page reports executed using personalized page execution mode into multiple individual pages of content. The first attribute on the page axis is used as the slicing attribute.

**smart fetch**
A type of report result fetching in which fetching continues until completion or you cancel the operation. This is the default fetch method used in MicroStrategy Office.

**smart tag alias**
A smart tag name that you can specify for a report or project. When Microsoft Office recognizes this name, the MicroStrategy Office Smart Tag actions menu is displayed. This means that when the name you specify is typed, it becomes a smart tag that provides several options to execute the report.

**Software Development Kit**
A distribution package of application program development software and the instructions for its use. Allows customization of an application.

**sort**
Arranging data according to some characteristic of the data itself (alphabetical descending, numeric ascending, and so forth). See also: drill, page-by, subtotal.

**sort by**
The order of the return values of an expression in relation to the order of the value or metadata object given. A sort by includes whether to sort in ascending or descending order, and which metadata object to sort by. Sort by may also be performed on the value of the subexpression, which is the input argument. To sort by a value or metadata object in an expression, you must set the SortBy parameter.

**SQL (Structured Query Language)**
The standardized query language established in 1986 by the American National Standards Institute (ANSI) and used to request information from tables in a relational database and to manipulate the tables' structure and data.

**SQL Engine**
The MicroStrategy Intelligence Server component that in report execution converts report requests into SQL to be used for a database query.
**static content**
Document content that is contained directly in the document and does not change from one service execution to the next. It cannot be personalized for different subscribers. Examples include an HTML template, a static URL, and so on. Compare with: dynamic content.

**Structured Query Language (SQL)**
A relational database language used to read data from tables in a relational database and to manipulate their structure and their data.

**subscriber**
A person who receives content from at least one service. Each individual who receives messages from Narrowcast Server has a login that provides password-controlled access to subscription and user preference information for the individual and his addresses. Subscribing at least one of these addresses to a particular service allows the individual to receive messages.

**subscription**
An enrollment in a service; a subscription is composed of one user and one address. subscription administrator Narrowcast Administrator user role. This role is designed for a console user who manages the Subscription Book, including users, addresses, and subscription sets. Also installs, configures, and administers the Subscription Book Module.

**Subscription Book**
Contains all users, addresses, and subscription sets. Stored in the Subscription Book Repository. See also: subscription set.

**Subscription Book Module**
A software component that provides the ability to create, edit, and use the objects within a Subscription Book. Those objects include users, addresses, and subscriptions. See also: Subscription Book Repository.

**Subscription Book Repository**
Stores all subscription information, including addresses and user information.

**Subscription ID**
The ID value used by Narrowcast Server to segment subscription sets. It is also called Segmentation ID. This value is required for dynamic subscription sets.

**subscription information**
All information related to an individual subscription. This information allows Narrowcast Server to deliver services to individual recipients in the desired manner.

**subscription information object**
Retrieves subscription information for a given segment. See also: dynamic subscription set, information object, segmentation, subscription information.
Subscription Portal
A feature of Narrowcast Server that allows end users to subscribe to Narrowcast Server services offered through a web-based portal. This enables end users to experience personalized and proactive interactions, based on user-defined permissions and preferences. See also: portal.

subscription set
A collection of addresses that can be subscribed to a service. Subscription sets are either static or dynamic. A static subscription set is an application object that retrieves subscription information from the Subscription Book Repository. Useful when the set of end user addresses that should receive a service does not change. A dynamic subscription set is an application object containing at least one piece of subscription information retrieved from an information object instead of from the Subscription Book Repository. Useful for changing or alert-driven subscription sets. Dynamic subscription sets can acquire some or all of their information from the information object. The rest of the subscription information is acquired from the Subscription Book Repository. See also: subscription set object.

Subscription Set ID
A numeric value used to identify each subscription set.

subscription set object
Application object that defines how to retrieve or modify a subscription set. Does not contain the subscription set, but provides instructions for how a Subscription Book Module should retrieve it. See also: Subscription Book Module, subscription set.

subtotal
A totaling operation performed for a portion of a result set. See also: drill and page-by.

summary metric
A shortcut to a subtotal, or a subtotal metric allowing explicit aggregation in documents. A summary metric allows you to select the function to use to calculate the subtotal.

system administrator
Narrowcast Administrator user role. This role is designed for a console user who performs the following tasks: Installs and configures the information delivery platform, Migrates system objects and application objects, Monitors, analyzes, and tunes the system to ensure the smooth and balanced operation of the decision support or business intelligence environment, Performs troubleshooting and error recovery.

system developer
Narrowcast Administrator user role. This role is designed for a Narrowcast Server user who employs the SDK and embeds Narrowcast Server technology into another product or application.
**system locale**
A language in which all objects are guaranteed to have a name. Since objects can have names in several locales (languages), it is necessary to have one locale where a name always exists for all objects. The system locale serves this role, because the system locale cannot be deleted. See also: locale, display locale, execution locale, selection locale, user locale, text container See: document (Narrowcast).

**table**
The primary physical component of a data warehouse, logically consisting of columns of data of varying types.

**template**
A MicroStrategy object that serves as a base on which you can build other objects of the same type. You can create a template for almost any kind of MicroStrategy object, such as filters or reports. Also see: Object template and Report template.

**text field**
A type of control in a document that displays text in the document. These different types of text content are: Static text, which does not change and serves as a label. Dynamic text, which is populated by the document or dataset. There are two types of dynamic text: Data field, which is populated from a dataset with data that originated in the data warehouse (or an Intelligence Server cache). A data field is only a reference to an object on a report. Auto text code, which is populated by the document or dataset, consisting of their settings rather than data from the data warehouse. A combination of any or all of the above types in one text field. See also: Data field, Auto text code.

**threshold**
Used to create conditional formatting for metric values. For example, a threshold triggers the report that, if dollar sales is greater than $200, format that cell to have a blue background with bold type.

**transformation**
A schema object that encapsulates a business rule used to compare results of different time periods. Transformations are used in the definition of a metric to alter the behavior of that metric.

**Unique Message Identifier**
An ID value that is unique for all messages delivered by any Narrowcast Server system. It can be used for message tracking purposes to determine which recipients have received and opened messages. This is typically done by creating a URL containing this ID value that refers to a zero-size image on a Web server that is configured to track references to this URL.
user hierarchy
  Named sets of attributes and their relationships, arranged in specific sequences for a logical business organization. They are user-defined and do not need to follow the logical model.

user ID
  A numeric value used to identify individual users.

user information
  The collection of information, including first name, last name, address, zip code, and other personal information, that changes from one subscriber to the next.

user locale
  Defines which content the user receives. This is set up by either the subscription administrator via Narrowcast Administrator or by the user via Subscription Portal. The user locale must match the selection locale. See also: locale, display locale, selection locale, system locale.

view filter
  The set of criteria that restricts the report data that is currently being viewed. It may include filtering conditions based on any of the objects on the report.

Visual Insight dashboard
  A visually-striking, interactive display that takes a minimal amount of time to set up and use. You can add text, interactive data visualizations, data filtering, and multiple layers of organization to your dashboard, then take advantage of Visual Insight’s formatting options to customize your display.

widget
  A type of control that presents data in a visual and interactive way; an interactive Flash-only graph that dynamically updates when a new set of data is selected. Some types include Gauge, Heat Map, and Stacked Area widgets.
INDEX

A
absolute path for microstrategy.xml_255
Access. See Microsoft Access_398
accessing
    MicroStrategy Connectivity Wizard_136
    MicroStrategy Installation Wizard
        UNIX_83
        Windows_54
    MicroStrategy Readme_134
    MicroStrategy Web Administrator page_218
        Oracle Glassfish Server_228
        Web JSP_206
activating installation_123
    activation code_121
    activating installation with_123
    requesting_121
    UNIX_92
    Windows_63
adding or removing MicroStrategy components (Windows)_377
admin role, assigning to a user
    JBoss - Windows - Web_252
    Oracle 10g - Windows - Web_247
    Oracle Glassfish Server_226
    Tomcat - Linux_240
    Tomcat - Windows - Web_235
    WebSphere - AIX - Web_218
    Administrative Server_214
AIX
    MicroStrategy Intelligence Server
        hardware requirements_25
        patch requirements_32
Amazon Redshift
    ODBC Driver for (Linux)_413
Amazon Redshift Wire Protocol
    ODBC Driver for_391
Analyst. See MicroStrategy Analyst_12
Analytics Modules. See MicroStrategy
    Analytics Modules_16
Apache Hive Wire Protocol
    ODBC Driver for_391
Apache Web server
    configuring to serve a static file_211
    deploying MicroStrategy Mobile
        Server JSP with_198
    deploying MicroStrategy Web JSP
        with_198
    proxy request to WebLogic_200
    starting_219
Architect. See MicroStrategy
    Architect_12
ASP.NET
    MicroStrategy Mobile Server and
        IIS_192,194
    MicroStrategy Web and IIS_191
Aster
    ODBC driver for (Linux)_410
attribute element_36
authentication_163
    anonymous_163
database_164
    Integrated_164
    LDAP_164
    OAuth for Data Import_255
    standard_163
    Windows_163
automatic deployment
    MicroStrategy Mobile Server
        JSP_201
    Web JSP_201
    WebLogic_201
B
backup frequency_38
C
    certified database and ODBC driver for
        MicroStrategy Intelligence
        Server_41
command line mode
    configuring metadata_359
    creating project source_370
MicroStrategy Configuration
    Wizard_358
    MicroStrategy Intelligence Server
        setup_366
Command Manager. See MicroStrategy
    Command Manager_9
common setups_19
    compatibility between Web JSP and
        Intelligence Server_41
component_1
    MicroStrategy Analytics Modules_16
    MicroStrategy Developer_12
    MicroStrategy Geospatial
        Information System (GIS)
        Connector_3
    MicroStrategy Health Center_17
    MicroStrategy Intelligence Server_6
    MicroStrategy Mobile_5
    MicroStrategy Narrowcast Server_11

© 2017, MicroStrategy Inc.
MicroStrategy Office_3
MicroStrategy Portlets_3
MicroStrategy SDK_14
MicroStrategy System Manager_11
MicroStrategy Tutorial Reporting_16
MicroStrategy Web_2
MicroStrategy Web Services_5
SequeLink ODBC Socket Server_11
Configuration Wizard. See MicroStrategy Configuration Wizard_143
configuring
   Apache Web server to serve a static file_211
   command line mode (UNIX)_357
   JBoss_250
   MicroStrategy command line mode_358
Oracle JDK
   JBoss on Windows_249
   Tomcat on Linux_238
   Tomcat on Windows_231
Tomcat
   Linux_238
   Windows_232
configuring ODBC
   UNIX_135
   Windows_134
connection errors, data source name_436
connection mapping_187
connection string_133
connectivity_138, 355
Connectivity Wizard. See MicroStrategy Connectivity Wizard_136

D

Data Import
   Dropbox configuration_255
   Facebook configuration_256
   Google Analytics configuration_256
   Google BigQuery configuration_256
   Google Drive configuration_256
   Salesforce configuration_256
   Twitter configuration_256
data source name_133
   connecting to_145
   connection errors_436
   creating with Connectivity Wizard_136
   creating with Microsoft ODBC Data Source Administrator_137
   data warehouse_181
data source, accessing_175
data warehouse_133
   communicating with_131
   connecting to_175
   connection setting_175
   DSN_181
   login_186
database authentication_164
database connection_181
certified ODBC driver_41
database instance_175
creating_180
database login_186
creating_186
database management system_(DBMS)_131
DataDirect Cloud
ODBC driver for (Linux)_412
DB Query Tool_139, 355
DB2 for i ODBC driver_392
DB2 for iSeries ODBC driver_392
DB2 ODBC driver (UNIX/Linux)_392
DB2 Wire Protocol ODBC driver
(Windows)_392
DB2 z/OS ODBC driver_393
DBMS. See database management system_131
deploying MicroStrategy Mobile Server
JSP_201
JBoss (Windows)_248
Oracle 10g_245
Oracle Glassfish Server_222-223
SAP NetWeaver (Windows)_242
Tomcat_230
WAR file with Oracle Glassfish
Server_225
WebLogic and Apache_198
WebLogic manual deployment_201
WebSphere and IBM HTTP
Server_212
deploying MicroStrategy Mobile Server
with ASP.NET and IIS_192
deploying MicroStrategy Operations
Manager with ASP.NET and IIS_194
deploying MicroStrategy Web_192
configuring data sources, Data
Import_255
deploying MicroStrategy Web JSP
JBoss (Windows)_248
Linux_236, 239
manually deploying with
WebLogic_201
Oracle 10g_245-246
Oracle Glassfish Server_222-223
SAP NetWeaver (Windows)_241-242
Tomcat_230, 236
Tomcat (Linux)_239
Tomcat (Windows)_233
WAR file with Oracle Glassfish
Server_225
WebLogic and Apache_198, 201
WebLogic automatic
deployment_201
WebSphere_213
WebSphere and IBM HTTP
Server_212
Windows_233
Developer. See MicroStrategy
Developer_12
direct project source_160
directory structure for UNIX_99
Distribution Services. See
MicroStrategy Distribution
Services_8
document_259
driver. See ODBC driver_41
Dropbox
configuring for Data Import_255
DSN Creator_436

E

Element browsing_36

Enterprise Manager. See MicroStrategy Enterprise Manager_10

Example

microstrategy.xml_255

response.ini file for MicroStrategy Configuration Wizard (Windows)_165

response.ini file for uninstallation_314

Excel. See Microsoft Excel_398

F

Facebook

configuring for Data Import_256

file

install.log_266

preinst.log_315

setup.exe_266

Folder structure for WebLogic and Apache_199

Freeform SQL, accessing data sources_175


G

Gateway_425

Google

Analytics, configuring for Data Import_256

BigQuery, configuring for Data Import_256

Drive, configuring for Data Import_256

Greenplum Wire Protocol

ODBC Driver for_397

H

Hadoop Gateway

caristics_426

deployment steps_427

prerequisites_426

system requirements and supported configurations_426

hardware recommendations

Web JSP on UNIX/Linux_25

Windows_22

hardware requirements for Intelligence Server on UNIX/Linux_25

Health Center. See MicroStrategy Health Center_17

History List

creating repository_149

tables_149

Hotfix

installation of MicroStrategy Office_318

silent installation (Windows)_317

HP-UX

MicroStrategy Intelligence Server hardware requirements_25

HTML document_258-259

setting up document creation_260

setting up document execution_263
IBM HTTP Server
configuring to serve a static file_222
IIS. See Microsoft IIS_34
Impala Wire Protocol
ODBC Driver for_393
Informix 8
ODBC driver for (UNIX/Linux)_406
Informix Wire Protocol
ODBC Driver for_394
install.log_266
installation
creating response.ini file
(Windows)_268
install.log file_266
OEM silent (Windows)_347
plug-in for Apache_200
reducing download size_18,46,54,84,339
response.ini file (Windows)_267
silent (UNIX)_321
silent (Windows)_315
silent installation of hotfix
(Windows)_317
troubleshooting silent
(Windows)_316
installation log file_266
installation method_45
Command line mode_45
GUI mode_45
silent installation_46
installation prerequisites_18
hardware recommendations for Web
JSP on UNIX/Linux_25
hardware recommendations for
Windows_22
hardware requirements for
Intelligence Server on
UNIX/Linux_25
large production deployment_21
recommended installation
location_19
small production deployment_20
software recommendations_29
software recommendations for SDK
with JSP_33
software recommendations for Web
JSP_33
software recommendations for Web
Services J2EE_33
software requirements for
Intelligence Server on
UNIX/Linux_29
installation procedure
checklist_48
Linux_79
MicroStrategy Installation Wizard
on Linux_83
MicroStrategy Installation Wizard
on UNIX_83
MicroStrategy Installation Wizard
on Windows_54
UNIX_79
Windows_53
Intelligence Server. See MicroStrategy
Intelligence Server_6
Intelligent Cube size_40
international support_41
Installation and Configuration Guide

J
Java heap size, setting
- Tomcat (Web) 233, 239
- WebLogic and Apache (Web) 209
- WebSphere (Web) 220

JBoss (Windows - Web)
- configuring 250
- preconfiguration information 249

JSP file, precompiling 209, 221

L
launching the project on WebLogic and Apache 208

launching WebSphere Administrative Console on Web 214

LDAP authentication 164

license 48

Linux
- deploying MicroStrategy Web JSP 236, 239
- installation procedure 79

MicroStrategy Installation Wizard 83

MicroStrategy Intelligence Server
- hardware requirements 25
- patch requirements 32

log
- installation 266
- statistics 38

M
manual deployment
- MicroStrategy Mobile Server JSP 201
- Web JSP 201

MDX cube source
- accessing a data source 175
- installing the MDX Cube Provider 66
- metadata 133
- table 146
- metadata repository 141, 146, 155, 345
- creating 145
- methods of automated installation (Windows) 266

Microsoft Access
- connecting as data source (UNIX/Linux) 398
- preparing database for access 398

Microsoft Excel
- connecting as data source 420
- connecting as data source (UNIX/Linux) 398
- creating DSN 421
- preparing as data source 421
- preparing file for access 399

Microsoft IIS 191
- ASP.NET and MicroStrategy Mobile Server 192, 194
- ASP.NET and MicroStrategy Web 191
- supporting IIS 7 or 7.5 34

Microsoft ODBC Data Source Administrator 137
- creating a DSN using 137

Microsoft SQL Azure
- ODBC driver 394
- Microsoft SQL Server ODBC
driver_394
MicroStrategy
   MDX Cube Provider_66
   uninstalling (UNIX/Linux)_382
MicroStrategy Analyst_12
MicroStrategy Analytics Modules_16
MicroStrategy Architect_12
MicroStrategy Command Manager_9
MicroStrategy common setups_19
MicroStrategy component_1
   MicroStrategy Analytics Modules_16
   MicroStrategy Developer_12
   MicroStrategy Geospatial Information System (GIS) Connector_3
MicroStrategy Health Center_17
MicroStrategy Intelligence Server_6
MicroStrategy Mobile_5
MicroStrategy Narrowcast Server_11
MicroStrategy Office_3
MicroStrategy Portlets_3
MicroStrategy SDK_14
MicroStrategy System Manager_11
MicroStrategy Tutorial Reporting_16
MicroStrategy Web_2
MicroStrategy Web Services_5
re-installing_378
re-installing MicroStrategy Office
   MicroStrategy Office
      re-installing_379
SequeLink ODBC Socket Server_11
MicroStrategy Configuration
   Wizard_142
accessing on UNIX/Linux_143
accessing on Windows_143
configuring MicroStrategy Health Center_145
connecting to a DSN_145
creating a metadata repository_145
prerequisites_141
response.ini file (Windows) 164,313
setting up Intelligence server_154
upgrading MicroStrategy_145
MicroStrategy Connectivity Wizard_136
accessing_136
creating a DSN_136
MicroStrategy Developer
   MicroStrategy Analyst_12
   MicroStrategy Architect_12
   MicroStrategy Function Plug-in Wizard_12
   MicroStrategy Server Administrator_13
      subcomponents_12
MicroStrategy Distribution Services_8
MicroStrategy Enterprise Manager_10, 151
MicroStrategy Function Plug-in Wizard_12
MicroStrategy Geospatial Information System (GIS) Connector_3
MicroStrategy Health Center_17
configuring_145
configuring with a response.ini file_164
MicroStrategy installation
Installation and Configuration Guide

prerequisites_18

MicroStrategy Installation Wizard
  Linux_83
  UNIX_83
  Windows_54

MicroStrategy Intelligence Server_6,
  154,157

MicroStrategy Distribution Services_8

MicroStrategy OLAP Services_7

MicroStrategy Report Services_7

MicroStrategy Transaction Services_9

MicroStrategy Mobile Server

MicroStrategy Mobile Server Administrator page

MicroStrategy Mobile Server JSP deployment_198

MicroStrategy Mobile support on IIS_7
  or 7.5_34

MicroStrategy Narrowcast Server_11

MicroStrategy Object Manager_14

MicroStrategy Office_3

MicroStrategy MDX Cube Provider
  installing_66

MicroStrategy Mobile_5

MicroStrategy OLAP Services_7
  sizing guidelines_37

MicroStrategy Mobile Server
  connecting to Intelligence Server_193-194

MicroStrategy Mobile Server
  deploying with ASP.NET and IIS_192

MicroStrategy Mobile Server
  WAR file location_195

MicroStrategy Mobile Server
  configuring for Oracle Glassfish Server_228

MicroStrategy Mobile Server
  configuring for WebSphere_218

MicroStrategy Mobile Server
  controlling access to
    JBoss Windows_252
    Tomcat Linux_240
    Tomcat Windows_235

MicroStrategy Mobile Server
  controlling access to (Oracle Glassfish Server)_226
MicroStrategy Operations Manager 11
developing with ASP.NET and IIS 194
MicroStrategy Operations Manager JSP
WAR file location 195
MicroStrategy Portlets 3
MicroStrategy Professional Services 38
MicroStrategy Readme, access to 134
MicroStrategy Report Services 7
setting up document execution 264
MicroStrategy Server Administrator 13
MicroStrategy System Manager
overview 11
MicroStrategy Transaction Services 9
MicroStrategy Tutorial Reporting 16
MicroStrategy Usher 17
MicroStrategy Web 190
allowing a user to install MicroStrategy Office from 57
ASP.NET 191
connecting to Intelligence Server 192
developing 192
supporting on IIS 7 or 7.5 34
versions 2
Web Analyst 2
Web Professional 2
Web Reporter 2
MicroStrategy Web Administrator
page 254
accessing (JBoss Windows) 253
accessing (Oracle) 247
accessing (Tomcat Linux) 241
accessing (Tomcat Windows) 235
controlling access to (JBoss Windows) 252
controlling access to (Oracle Glassfish Server) 226
controlling access to (Tomcat Linux) 240
controlling access to (Tomcat Windows) 235
MicroStrategy Web JSP
compatibility with Intelligence Server 41
connection 254
developing automatically with WebLogic 201
developing manually with WebLogic 201
deploying on Oracle 10g 246
deploying with JBoss (Windows) 248
deploying with SAP NetWeaver (Windows) 241
hardware recommendations for UNIX/Linux 25
launching project (JBoss Windows) 253
launching project (Oracle 10g) 248
launching project (Tomcat Windows) 236
software recommendations 33
undeploying 230
WAR file location 195
MicroStrategy Web Universal
Web Reporter 2
microstrategy.xml
absolute path 255
example_255
sample modified_255
migrating MicroStrategy Intelligence Server_97
MongoDB
   ODBC Driver for_395
MSTRInst.log_315
MultiSource Option_9_175
MySQL
   ODBC driver for (Linux)_409
   N
Narrowcast Server. See MicroStrategy Narrowcast Server_11
Netezza
   ODBC driver for (UNIX/Linux)_407
   network bandwidth_40
   O
Object Manager. See MicroStrategy Object Manager_14
ODBC_131_133
   default directory (UNIX)_135
   default directory (Windows)_134
ODBC connectivity_138_355
ODBC driver_133
   Amazon Redshift (Linux)_413
   Amazon Redshift Wire Protocol_391
   Apache Hive Wire Protocol_391
   Aster (Linux)_410
certified_134
certified database_41
DataDirect Cloud (Linux)_412
DB Query Tool_139_355
DB2 (UNIX/Linux)_392
DB2 (Windows)_392
DB2 for i_392
DB2 for iSeries_392
DB2 z/OS_393
Greenplum Wire Protocol_397
Impala Wire Protocol_393
Informix 8 (UNIX/Linux)_406
Informix Wire Protocol_394
Microsoft SQL Azure_394
Microsoft SQL Server_394
MongoDB_395
MySQL (Linux)_409
Netezza (UNIX/Linux)_407
odbc.ini file (UNIX/Linux)_424
Oracle Wire Protocol_396
other relational databases for Windows_420
PostgreSQL Wire Protocol_397
Red Brick (Linux)_402
Salesforce_402
SAP HANA_416
SequeLink_398
Sybase Adaptive Server IQ (UNIX/Linux)_403
Sybase ASE Wire Protocol_397
Teradata (UNIX/Linux)_405
Teradata (Windows)_420
testing connectivity_138_355
Vertica (Linux)_415
odbc.ini_424
OLAP Services. See MicroStrategy OLAP Services_7
Open Database Connectivity_131
Operations Manager. See MicroStrategy Operations Manager_11
options file_321
Oracle 10g
deploying MicroStrategy Mobile Server JSP_245
deploying MicroStrategy Web JSP_245-246
deploying Web JSP with the Oracle Enterprise Manager_246
preconfiguration information (Web)_245
Oracle Glassfish Server
accessing MicroStrategy Web Administrator page_228
connecting to the Project page_230
deploying MicroStrategy Mobile Server JSP_222-223
deploying MicroStrategy Mobile Server JSP as a WAR file_225
deploying MicroStrategy Web JSP_222-223
deploying MicroStrategy Web JSP as a WAR file_225
recommended configuration for UNIX/Linux_222
undeploying Mobile Server JSP_230
undeploying Web JSP_230
Oracle Glassfish Server (Web)
preconfiguration information_222
recommended configuration for UNIX/Linux_222
Oracle Glassfish Server Admin Console launching_224
Oracle JDK
configuring_231,249
Oracle Wire Protocol
ODBC driver for_396
pageCheckSeconds parameter_210
password to metadata_155
patch requirements
AIX_32
Linux_32
Solaris_32
performance-based setup
MicroStrategy Professional Services_38
WebLogic and Apache (Web)_208
permission errors_437
plug-in for Apache_200
plugin-cfg.xml, regenerating_217
port number_162
Intelligence Server_157
portlets. See MicroStrategy Portlets_3
PostgreSQL Wire Protocol
ODBC driver for_397
precompiling
JSP file in WebLogic_209
JSP file in WebSphere_221
preconfiguration information
Apache (Web)_199
JBoss (Windows - Web)_249
Oracle 10g (Web)_245
Oracle Glassfish Server (Web)_222
Installation and Configuration Guide

Tomcat (Linux - Web) 237
Tomcat (Windows - Web) 230
WebLogic (Web) 199
WebSphere (Web) 213
preinst.log 315
prerequisites for MicroStrategy Configuration Wizard 141
processor speed 39
production deployment
  large 21
  small 20
project 175
  accessing multiple data sources 175
project source 144, 160
  authentication 163
  direct 160
  server 160
  three-tier 161
  two-tier 160
Q
Query Builder, accessing data sources 175
R
R (analytics)
  integrating with MicroStrategy 12
R Integration Pack 12
re-deploying Web JSP with WebLogic 208
re-installing a MicroStrategy component (Windows) 378
  control panel 378
re-installing MicroStrategy Office 379
Red Brick
  ODBC driver for 402
requesting activation code 121
response.ini file
  creating for installation (Windows) 268
  creating through MicroStrategy Configuration Wizard (Windows) 165
example for uninstall (Windows) 314
installation (Windows) 267
MicroStrategy Configuration Wizard (Windows) 164, 313
MicroStrategy Health Center 164
parameters and options for Configuration Wizard (Windows) 167
parameters and options for installation (Windows) 268
uninstalling (Windows) 313
upgrade 164
  using for Configuration Wizard (Windows) 166
  using for installation (Windows) 312
  using through Configuration Wizard (UNIX) 371
restarting WebSphere application server
  Web 217
S
Salesforce
  configuring for Data Import 256

© 2017, MicroStrategy Inc.
Installation and Configuration Guide

ODBC driver for SAP HANA

ODBC driver for SAP NetWeaver (Windows)

deploying MicroStrategy Web JSP

SDK. See MicroStrategy SDK

security

access to MicroStrategy Web Administrator page

SequeLink

configuring ODBC driver

ODBC driver for SequeLink ODBC Socket Server

Server Administrator. See MicroStrategy Server Administrator

server definition

connectivity setting

server project source

setting up document creation

setting up document execution

HTML document

setup command-line parameter (Windows)

setup.exe

silent installation

hotfix (Windows)

introduction (UNIX)

OEM (Windows)

options file (UNIX)

output (UNIX)

output (Windows)

parameters (UNIX)

perform on Windows

troubleshooting (Windows)

values for different languages

sizing guidelines

caching

Intelligent Cube size

MicroStrategy OLAP Services

MicroStrategy Professional Services

network bandwidth

number of users

processor speed

report complexity

Report Services document

software recommendations

SDK for JSP

Web JSP

Web Services J2EE

software requirements for Intelligence Server on UNIX/Linux

Solaris

MicroStrategy Intelligence Server hardware requirements

patch requirements

specifying a port number

SQL

query

script

© 2017, MicroStrategy Inc.
starting
Administrative Server 214
Apache Web Server 219
Web module for WebSphere 218
WebSphere Administrative Console 214
WebSphere default application server 215
static file
configuring Apache Web server for 211
configuring IBM HTTP Server 222
statistics
log 38
table 151
support
international 41
Sybase Adaptive Server IQ
ODBC driver for (UNIX/Linux) 403
Sybase ASE Wire Protocol
ODBC driver for 397
System Manager. See MicroStrategy System Manager 11
T
table, metadata 146
Teradata
ODBC driver for (UNIX/Linux) 405
ODBC driver for (Windows) 420
three-tier project source 161
Tomcat
deploying MicroStrategy Mobile Server JSP 230
deploying MicroStrategy Web JSP 230, 236
Tomcat (Linux - Web)
deploying MicroStrategy Web JSP 239
preconfiguration information 237
setting Java heap size 239
Tomcat (Web)
configuring 232, 238
configuring Oracle JDK 238
Tomcat (Windows - Web)
preconfiguration information 230
setting Java heap size 233
Tomcat (Windows)
deploying MicroStrategy Web JSP 233
Transaction Services. See MicroStrategy Transaction Services 9
troubleshooting
DSN connection errors 435
History List creation errors 436
metadata creation errors 436
permission errors 437
server port number errors 435
statistics creation errors 436
UNIX 434
Tutorial Reporting. See MicroStrategy Tutorial Reporting 16
Twitter
configuring for Data Import 256
two-tier project source 160
unattended installation (UNIX) 321
uninstalling
Control Panel (Windows)_380
MicroStrategy (UNIX/Linux)_382
response.ini (Windows)_313
uninstalling MicroStrategy components (Windows)_380
MicroStrategy Health Center (Windows)_381
MicroStrategy Office (Windows)_381
UNIX
configuring ODBC_135
installation procedure_79
MicroStrategy Installation Wizard_83
verifying installation_98
user assigned to admin role
JBoss - Windows (Web)_252
Oracle 10g - Windows (Web)_247
Oracle Glassfish Server_226
Tomcat - Linux_240
Tomcat - Windows (Web)_235
WebSphere - AIX- Web_218
user connection mapping. See connection mapping_187
Usher. See MicroStrategy Usher_17

V
verifying installation
UNIX_98
Windows_68
Vertica
ODBC driver for (Linux)_415

W
WAR file
WAR file (Web)
exploding for WebLogic and Apache (Web)_203
unzipping for WebLogic and Apache (Web)_205
Web JSP. See also MicroStrategy Web JSP_2
Web module for WebSphere,
starting_218
Web server OS
AIX_33
HP-UX_33
Linux_33
Windows_33
Web. See MicroStrategy Web_2
web.xml
re-deploying the application_208
WebLogic (Solaris - Web)_198
WebLogic and Apache
deploying MicroStrategy Mobile Server JSP_199
production mode_201
WAR file_205
WebLogic and Apache (Web)
deploying MicroStrategy Web JSP_198_201
exploding the WAR file_203
folder structure_199
launching the project_208
pageCheckSeconds parameter_210
performance-based setup_208
precompiling JSP file_209
preconfiguration information_199
setting Java heap size_209
WebSphere (Web)_212
  application installation_215
  deploying MicroStrategy Web_JSP_213
  launching Administrative Console_214
  precompiling the JSP file_221
  preconfiguration_213
  regenerating plugin-cfg.xml_217
  restarting application server_217
  setting Java heap size_220
  starting default application server_215
WebSphere and IBM HTTP Server
  deploying MicroStrategy Web_JSP_212
Windows
  authentication_163
  configuring ODBC_134
  deploying MicroStrategy Web_JSP_233
  hardware recommendations_22
  installation procedure_53
  MicroStrategy Installation Wizard_54
  verifying installation_68
  Web server OS_33
  Windows INI_267

X
XSL report style_38

© 2017, MicroStrategy Inc.