

Installation and Configuration Guide

10.8, March 2018

Copyright © 2018 by MicroStrategy Incorporated. All rights reserved.

Trademark Information

The following are either trademarks or registered trademarks of MicroStrategy Incorporated or its affiliates in the United States and certain other countries:

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

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

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

Patent Information

This product is patented. One or more of the following patents may apply to the product sold herein: U.S. Patent Nos. 6,154,766, 6,173,310, 6,260,050, 6,263,051, 6,269,393, 6,279,033, 6,567,796, 6,587,547, 6,606,596, 6,658,093, 6,658,432, 6,662,195, 6,671,715, 6,691,100, 6,694,316, 6,697,808, 6,704,723, 6,741,980, 6,765,997, 6,768,788, 6,772,137, 6,788,768, 6,798,867, 6,801,910, 6,820,073, 6,829,334, 6,836,537, 6,850,603, 6,859,798, 6,873,693, 6,885,734, 6,940,953, 6,964,012, 6,977,992, 6,996,568, 6,996,569, 7,003,512, 7,010,518, 7,016,480, 7,020,251, 7,039,165, 7,082,422, 7,113,993, 7,127,403, 7,174,349, 7,181,417, 7,194,457, 7,197,461, 7,228,303, 7,260,577, 7,266,181, 7,272,212, 7,302,639, 7,324,942, 7,330,847, 7,340,040, 7,356,758, 7,356,840, 7,415,438, 7,428,302, 7,430,562, 7,440,898, 7,486,780, 7,509,671, 7,516,181, 7,559,048, 7,574,376, 7,617,201, 7,725,811, 7,801,967, 7,836,178, 7,861,161, 7,861,253, 7,881,443, 7,925,616, 7,945,584, 7,970,782, 8,005,870, 8,051,168, 8,051,369, 8,094,788, 8,130,918, 8,296,287, 8,321,411, 8,452,755, 8,521,733, 8,522,192, 8,577,902, 8,606,813, 8,607,138, 8,645,313, 8,761,659, 8,775,807, 8,782,083, 8,812,490, 8,832,588, 8,943,044, 8,943,187, 8,958,537, 8,966,597, 8,983,440, 8,984,274, 8,984,288, 8,995,628, 9,027,099, 9,027,105, 9,037, 577, 9,038,152, 9,076,006, 9,086,837, 9,116,954, 9,124,630, 9,154,303, 9,154,486, 9,160,727, 9,166,986, 9,171,073, 9,172,699, 9,173,101, 9,183, 317, 9,195,814, 9,208,213, 9,208,444, 9,262,481, 9,264,415, 9,264,480, 9,269,358, 9,275,127, 9,292,571, 9,300,646, 9,311,683, 9,313,206, 9,330,174, 9,338,157, 9,361,392, 9,378,386, 9,386,416, 9,391,782, 9,397,838, 9,397,980, 9,405,804, 9,413,710, 9,413,794, 9,430,629, 9,432,808, 9,438,597, 9,444,805, 9,450,942, 9,450,958, 9,454,594, 9,507,755, 9,513,770, 9,516,018, 9,529,850, 9,563,761, 9,565,175, 9,608,970, 9,640,001, 9,646,165, 9,680,908, 9,697,146, 9,697,350, 9,742,764, 9,742,781, and 9,743,235. Other patent applications are pending.

CONTENTS

Book Overview and Additional Resources	7
About this book	7
1. Planning Your Installation	19
Upgrade considerations	19
MicroStrategy products and components	19
Installation prerequisites	50
Installation considerations	65
Methods of installation	76
Licensing information	79
Installation and configuration checklists	79
2. Installing MicroStrategy on Windows	82
Installation procedure in Windows	83
Configuring your MicroStrategy installation	107
3. Installing MicroStrategy on Linux	109
Installation procedures on Linux	109
Configuring your MicroStrategy installation	129
4. Installing and Configuring Usher	130
Overview of Usher Install	130
Usher Pre-Installation Instructions	131
Usher Installation Instructions	138
Usher Post-Installation Instructions	138
Troubleshooting Information	144
5. Activating Your Installation	148
Request an Activation Code	148

Activate your installation	150
Configuring your MicroStrategy installation	151
Server Activation FAQ	151
6. Configuring and Connecting Intelligence Server	157
Communicating with databases	157
Initial MicroStrategy configuration	166
Connecting to a data warehouse and other repositories	199
Creating a project	212
Configuring your MicroStrategy installation	212
7. Deploying MicroStrategy Web and Mobile Server	214
Deploying with IIS (Windows)	215
General steps to deploy MicroStrategy JSP applications	218
Deploying with WebLogic and Apache (Solaris)	221
Deploying with WebSphere and IBM HTTP Server (AIX)	234
Deploying with Oracle Glassfish Server (Solaris)	243
Deploying with Tomcat (Windows)	250
Deploying with Tomcat (Linux)	255
Deploying with SAP NetWeaver (Windows)	259
Deploying with Oracle 10g (Windows)	262
Deploying with JBoss (Windows)	265
Administering your MicroStrategy Web deployment	269
Using absolute paths to share configuration files	271
Configuring third-party data sources for importing data	271
Configuring your MicroStrategy installation	273
8. Setting Up Documents and HTML Documents	274
Prerequisites	274
Executing documents and HTML documents in Linux	275
Configuring your MicroStrategy installation	280
9. Automated Installation on Windows	281
Installation log file	282
Methods of installation	282
Installing and configuring with a response.ini file	283
Silent installation	324
Configuring your MicroStrategy installation	329

10. Automated Installation on Linux	330
Silent installation	330
Configuring MicroStrategy in command line mode	349
Configuring your MicroStrategy installation	349
11. Deploying OEM Applications	350
Creating DSNs for OEM environments	351
Configuring a MicroStrategy installation	352
Designing a project and reporting environment	352
Customizing MicroStrategy Web	353
Deploying a MicroStrategy OEM application	353
Tuning an OEM deployment	358
Updating OEM applications	359
Troubleshooting support for MicroStrategy OEM applications	360
12. Configuring MicroStrategy Using Command Line Tools	361
Creating a DSN for a data source	361
Testing ODBC connectivity	362
Configuring MicroStrategy with a response.ini file	364
Configuring and controlling Intelligence Server	378
Supporting reserved words and characters	382
Configuring your MicroStrategy installation	382
13. Adding or Removing MicroStrategy Components	383
Adding or removing MicroStrategy components on Windows	383
Re-installing MicroStrategy components on Windows	384
Uninstalling MicroStrategy components on Windows	385
Uninstalling MicroStrategy components on Linux	387
14. Export Engine configuration	389
Installation of Export Engine	389
Changing the port of the Export Engine	389
Connecting Intelligence Server to a specific Export Engine	390
Connect the Export Engine to a specific Kafka server	390
Increase concurrency of Export Engine	391
A.Connecting to Databases and Data Sources	392
Creating DSNs for specific data sources	392
Creating database connections in Web	426

Configuring ODBC parameters with odbc.ini	428
B.Installing MicroStrategy Hadoop Gateway	430
Connection Modes	430
Constraints	431
System requirements and supported configurations	431
Prerequisites	431
Steps to deploy MicroStrategy Hadoop Gateway	432
How to manually deploy Hadoop Gateway if the Hadoop cluster has Kerberos authentication enabled	440
Import Data from Hadoop	441
C.Troubleshooting	442
Reviewing general installation errors	442
Graph and document support of non-Western European fonts	442
Server port number errors	443
DSN connection errors	443
Metadata and other repository creation errors	444
Permission errors	445
D. Usher Administration	446
Managing Usher Administrators	446
Managing the Usher Signing Certificate Authority	447

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 Linux and Windows versions of the MicroStrategy platform.

For Linux installations, this guide assumes a basic understanding of how to use Linux either from a 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 79 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 used 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 10.9

- On Linux environments, all MicroStrategy processes will now be registered as OS services upon restart once installation is complete. For a list of the processes registered, see [Installation procedures on Linux](#).
- For Linux installations, the GUI now provides an option to register MicroStrategy processes as a service. If you are installing as a non-root user the location of sample files provided to register the services after installation.

MicroStrategy 10.8

- R and The R Integration Pack can now be installed and configured automatically when installing your MicroStrategy platform. R is used by Intelligence Server to process 'R' based functions to enable 'R' analytics.
- MicroStrategy Operations Manager will no longer be included with new installations or upgrades. Enterprise Manager functionality is now controlled exclusively through Command Manager.

MicroStrategy 10.7

- For Windows installations using the Installation Wizard, an option has been added to the Review Installation Settings window to automatically restart your computer when the

installation is complete.

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](#).
- Windows environments now include MicroStrategy Messaging Services. For more information, see *Messaging Services is a component that is coupled with the Intelligence Server during installations and upgrades. Messaging Services is configured out-of-the-box and runs automatically after the installation is completed. Logs will be sent to MicroStrategy Messaging Services Server only when Messaging Services feature is enabled and Kafka Server can be connected successfully. Logs will be sent to local disk if Messaging Services is disabled or the Kafka Server is down or unreachable because of network issues.*

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 *Messaging Services is a component that is coupled with the Intelligence Server during installations and upgrades. Messaging Services is configured out-of-the-box and runs automatically after the installation is completed. Logs will be sent to MicroStrategy Messaging Services Server only when Messaging Services feature is enabled and Kafka Server can be connected successfully. Logs will be sent to local disk if Messaging Services is disabled or the Kafka Server is down or unreachable because of network issues.*
- Beginning with MicroStrategy 10.5, Operations Manager will no longer be included with new installations. If you are upgrading from a previous version and have Operations Manager installed and activated, it will be available as an option. New installations will administer Enterprise Manager by using Command Manager.
- 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 [Installing and Configuring Usher](#).

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 7, Deploying MicroStrategy Web and Mobile Server](#).
- 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 271](#).
- 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 426](#).

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 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](#).
- Information on configuring the MicroStrategy ODBC Driver for Impala Wire Protocol is provided in [MicroStrategy ODBC Driver for Impala Wire Protocol for Windows and Linux, page 398](#).

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

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

To access documentation resources from any location, [click here](#).

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>. For example, type “MicroStrategy derived metric” or “MicroStrategy logical table” into a Google search. As described above, books typically describe general concepts and examples; Help typically provides detailed steps and screen options. To limit your search to MicroStrategy books, on Google’s main page you can click **More**, then select **Books**.

Additional formats

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

For iOS devices, scan the following QR code:



For Android devices, scan the following QR code:



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

Guides for MicroStrategy overview and evaluation

- *Introduction to MicroStrategy*

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***
- ***MicroStrategy Report Services Document Creation Guide***

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***

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***

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***

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***

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

- ***MicroStrategy Supplemental Reference for System Administration***

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.

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 [here](#) 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 [here](#).

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

To access documentation resources on Windows

- 1 In Windows, choose **Start > Programs (or All Programs) > MicroStrategy Documentation > Product Manuals**.
- 2 Click the link for the desired manual or other documentation source.
- 3 If you click the link for the Narrowcast Services SDK Guide, a File Download dialog box opens. This documentation resource must be downloaded. Select **Open this file from its current location**, and click **OK**.



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

To access documentation resources on UNIX and Linux

- 1 Within your UNIX or Linux machine, navigate to the directory where you installed MicroStrategy. The default location is `/opt/MicroStrategy`, or `$HOME/MicroStrategy/install` if you do not have write access to `/opt/MicroStrategy`.
- 2 From the MicroStrategy installation directory, open the `Help` folder.
- 3 Open the `Product Manuals.htm` file in a web browser. A page opens in your browser showing a list of available manuals in PDF format and other documentation sources.
- 4 Click the link for the desired manual or other documentation source.

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





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

Documentation standards

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



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

Type	Indicates
bold	<ul style="list-style-type: none"> 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 <p>Example: Click Select Warehouse.</p>
<i>italic</i>	<ul style="list-style-type: none"> Names of other product manuals and documentation resources When part of a command syntax, indicates variable information to be replaced by the user <p>Example: The <i>aggregation level</i> is the level of calculation for the metric.</p> <p>Example: Type <code>copy c:\filename d:\foldername\filename</code></p>
Courier font	<ul style="list-style-type: none"> Calculations Code samples Registry keys Path and file names URLs Messages displayed in the screen Text to be entered by the user <p>Example: <code>Sum(revenue)/number of months</code>.</p> <p>Example: Type <code>cmdmgr -f scriptfile.scp</code> and press Enter.</p>
+	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.

PLANNING YOUR INSTALLATION

MicroStrategy business intelligence tools help organizations to monitor, report, and analyze all of their enterprise data. MicroStrategy helps you make decisions based upon the data within your organization's enterprise data warehouses and other business data sources.

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



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

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

Upgrade considerations

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

MicroStrategy products and components

MicroStrategy has a range of products and components that you can install on different operating systems. Depending on the type of setup that you have, you can install various combinations of MicroStrategy components. The components described in this section offer a complete set of tools for creating, deploying, supporting, and maintaining your business intelligence applications.

MicroStrategy components and their subcomponents are described in relation to how the components are grouped together during the installation routine, as well as how they fit into MicroStrategy's product offerings.

Memory Allocation for MicroStrategy Products and Services

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

Intelligence Server

Recommended Memory	Component	Services	Memory at Rest
8 GB	Intelligence Server Module	MSTRSvr2_64 (MicroStrategy Intelligence Server)	50 MB
		The non-sucking service manager (MicroStrategy Intelligence Server Log Consumer)	2 MB
		Java™ Platform SE binary	500 MB
		The non-sucking service manager (MicroStrategy PDFExport Service)	2 MB
		Java™ Platform SE binary	525 MB
		MJRefSvr_64	100 MB
		MJHGoSMgr_64	1GB
	Universal Option		
	Report Services		
	OLAP Services		
	Distribution Services		
	Transaction Services		
	Multisource Option		
	Clustering Option		

MicroStrategy Web

Recommended Memory	Component	Services	Memory at Rest
7 GB	Web Server (ASP.NET)	IIS Worker Process	500 MB
	Web Server (JSP)		
	Web Universal Analyst		
	Web Universal Professional		
	Web Universal Reporter		
	Portlets		
	GIS Connectors		

MicroStrategy Office

Recommended Memory	Component	Services	Memory at Rest
3 GB	Office Client	moipkg (Office Package Wizard)	10 MB
		moicnfg (Office Configuration)	30 MB
	Web Services for Office (ASP.NET)	IIS Worker Process	100 MB
	Web Services for Office (JSP)		

MicroStrategy Mobile

Recommended Memory	Component	Services	Memory at Rest
2 GB	Mobile Server(ASP.NET)	IIS Worker Process	500 MB
	Mobile Server(JSP)		

MicroStrategy Developer

Recommended Memory	Component	Services	Memory at Rest
5 GB		MSTRDesk	40 MB
		MicroStrategy.XEG.WPFApp (Xquery Editor and Generator)	50 MB
	Analyst		
	Developer		
	Architect	MSTRDesk (MicroStrategy Architect - <Project Name>)	
	Architect Function Plugin		
	Server Administrator		

MicroStrategy Messaging Services

Recommended Memory	Services	Memory at Rest
1 GB	Java™ Platform SE Binary (Apache Zookeeper)	75 MB
	Java™ Platform SE Binary	100 MB
	Java™ Platform SE Binary (Apache Kafka)	75 MB
	Java™ Platform SE Binary	500 MB

MicroStrategy Object Manager

Recommended Memory	Services	Memory at Rest
1 GB	ObjectManager	25 MB
	ProjectMergeUI	25 MB
	MARTT2UI	20 MB
	MergeUtility	20 MB

Command Manager

Recommended Memory	Services	Memory at Rest
1 GB	CmdMgrW	65 MB

Enterprise Manager

Recommended Memory	Services	Memory at Rest
1 GB	MJMulPrc_64	10 MB
	MSTREMSservice, MJMulPrc_64 (4) (EM Service Running)	50 MB

Narrowcast Server

Recommended Memory	Component	Services	Memory at Rest
5 GB	Narrowcast Administrator	MSTRNCAD (MicroStrategy Narrowcast Administrator)	25 MB
		Monitor (Monitor)	1 MB
		MicroStrategy Logging Client	
		MicroStrategy System Monitor	
	Delivery Engine	MicroStrategy Distribution Manager	
		MicroStrategy Execution Engine	
		MicroStrategy Logging Consumer	
		MicroStrategy Logging Server	
		MicroStrategy NC PDF Formatter	
		MicroStrategy SMTP Service	
	Subscription Portal	Portal Administrator	
		Subscription Portal	
	Tutorial Delivery Installation		
	Tutorial Delivery Configuration		

Analytics Module

Recommended Memory	Component	Services	Memory at Rest
7 GB	N/A	N/A	N/A

Other Components

Recommended Memory	Component	Services	Memory at Rest
4 GB	SequeLink ODBC Socket Server	DataDirect SequeLink Agent (SLAgent55)	2 MB
		DataDirect SequeLink Server for ODBC Socket (SLSocket55)	2 MB
		DataDirect SequeLink Service Starter	1 MB
		DataDirect Technologies Home	
		ipexen (DataDirect Product Registration)	
		Microsoft Management Console (sladmin60)	
		Windows Command Processor (SequeLink Management Command Line Tool)	
	MDX Cube Provider		
	MySQL	mysqld	450 MB
	Apache Tomcat	Commons Daemon Service Runner (Apache Tomcat 8.0 Tomcat8)	300 MB

Mandatory tools installed with each MicroStrategy Product

Component	Services	Memory at Rest
Health Agent	MSTRExec (MicroStrategy Health Agent)	50 MB
Health Centre Console	MFDgnVwr (MicroStrategy Health Centre Console)	50 MB
Configuration Wizard	macfgwizw (Configuration Wizard)	50 MB

Component	Services	Memory at Rest
Diagnostics and Performance Monitoring Tool	MADPCfg (MicroStrategy Connectivity Wizard)	50 MB
License Manager	MALicMgrW_64 (MicroStrategy License Manager)	75 MB
Connectivity Wizard	MAMDCW (MicroStrategy Connectivity Wizard)	75 MB
Project Source Manager (Everything except Web, Mobile, Office)	MACONMAN	15 MB

Tools installed with some MicroStrategy Products

Component	Services	Memory at Rest
Project Source Manager (Everything except Web, Mobile, Office)	MACONMAN	15 MB
Listener Service (Intelligence Server, Enterprise Manager)	MSTRLSn2_64 (MicroStrategy Listener)	10 MB
Test Listener (Everything except Web, Mobile, Office)	TestListener (MicroStrategy Test Listener)	10 MB
DB Query Tool (Everything except Web, Mobile, Office)	MADBQueryTool (MicroStrategy DB Query Tool)	50 MB
Cube Advisor (Everything except Web, Mobile, Office)	MstrCubeAdvisor (MicroStrategy Cube Advisor)	15 MB
Service Manager (Intelligence Server, Enterprise Manager, Narrowcast)	MASvcMgr_64 (MicroStrategy Service Manager)	80 MB

MicroStrategy Web

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

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

MicroStrategy Web components

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

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

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

For information about configuring and tuning MicroStrategy Web, refer to the [System Administration Guide](#).

MicroStrategy Web versions

MicroStrategy Web is available in the following versions:

- **Web Reporter:** Business users are able to view all types of reports and scorecards and also personalize reports, print, drill, sort, export, choose between grid or graph format, and schedule or immediately send reports via email or to a file server or a printer.
- **Web Analyst:** This version provides all the functionality of Web Reporter plus the ability to drill anywhere, edit totals, pivot reports, add or remove fields from reports, create derived metrics, and create reports or ad hoc queries.
- **Web Professional:** This full-featured version provides all the functionality of Web Analyst plus the ability to design scorecards, dashboards, and operational reports in design mode or WYSIWYG view mode. Web Professional users have advanced formatting capabilities as well as the ability to perform calculations across multiple data sources.

MicroStrategy Portlets

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

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

In portal environments, users are commonly already logged in and authenticated with the portal. This authentication can also be used to provide access to MicroStrategy Web within the portal without having to re-enter their login information. This process is known as single sign-on. Out-of-the-box MicroStrategy Portlets automatically include support for single sign-on.

For steps to install and configure out-of-the-box MicroStrategy Portlets for several major enterprise portal servers, see the *Enterprise Portal Integration Help*. This resource can be accessed from the MicroStrategy Product Manuals page, as described in [Accessing manuals and other documentation sources, page xxiii](#).

MicroStrategy GIS Connectors

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

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

MicroStrategy Office

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

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

To learn how to use MicroStrategy Office, refer to the [MicroStrategy Office User Guide](#) and MicroStrategy Office online help.



MicroStrategy Office requires that MicroStrategy Web Services is also installed. For information on Web Services, see [MicroStrategy Web Services \(ASP.NET\) and Web Services \(J2EE\), page 28](#).

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 327).

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

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

- MicroStrategy Web Services (ASP.NET) is an easy-to-deploy service. You can deploy the ASP.NET version using Microsoft IIS on a Windows environment.
- MicroStrategy Web Services (J2EE) provides a servlet-based version of MicroStrategy Web Services that is compatible with a Linux or Windows environment.

To support alternative ways to access the MicroStrategy business intelligence platform using the latest web services technologies such as ASP.NET, JNI, Java and Web protocols, such as Apache Axis, refer to the [MicroStrategy SDK](#), page 37 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 7, Deploying MicroStrategy Web and Mobile Server](#).

MicroStrategy Server

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

- 64-bit server infrastructure to scale to big data volumes and a large number of users.
- Ability to connect to and join data from multiple data sources.
- In-memory acceleration of analytical processing for instantaneous response.
- Processing of all analytic styles from self-service data discovery to beautiful, immersive information apps to the industry's broadest spectrum of advanced analytics.
- Proactive distribution of personalized reports and alerts.
- Ability to embed actionable intelligence in analytical applications.

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

MicroStrategy Intelligence Server

MicroStrategy Intelligence Server delivers world-class monitoring, reporting, and analysis on a single integrated platform, offering next generation BI capabilities for the full range of BI applications. MicroStrategy Intelligence Server is the architectural foundation of the MicroStrategy platform. It performs the following critical tasks for the MicroStrategy BI platform:

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

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

All other products in the platform work in conjunction with Intelligence Server and benefit from its broad functionality.

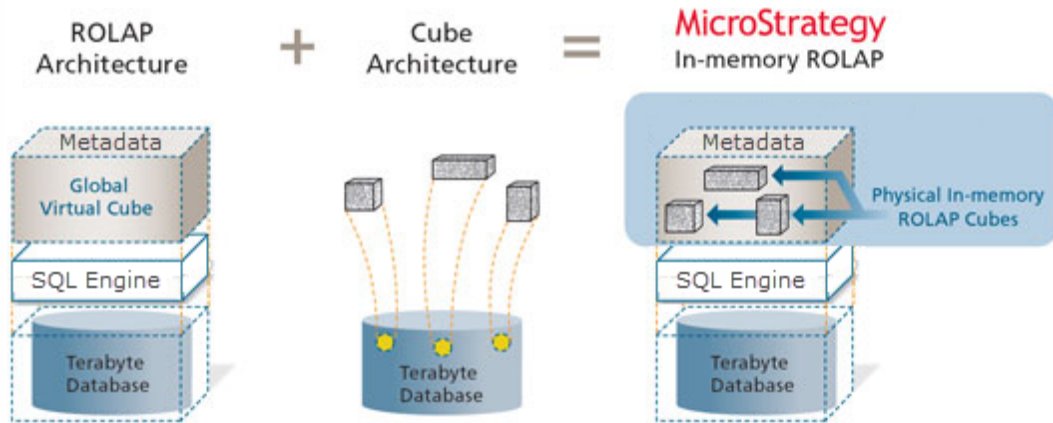
The subcomponents of MicroStrategy Intelligence Server are as follows:

- [MicroStrategy OLAP Services, page 30](#)
- [MicroStrategy Report Services, page 31](#)
- [MicroStrategy Distribution Services, page 32](#)
- [MicroStrategy Transaction Services, page 32](#)
- [MultiSource Option, page 32](#)
- Clustering Option, which allows you to cluster a group of Intelligence Server machines (up to four Intelligence Server machines) to take advantage of the many benefits available in a clustered environment.

For information on clustering Intelligence Servers, see the [System Administration Guide](#).

MicroStrategy OLAP Services

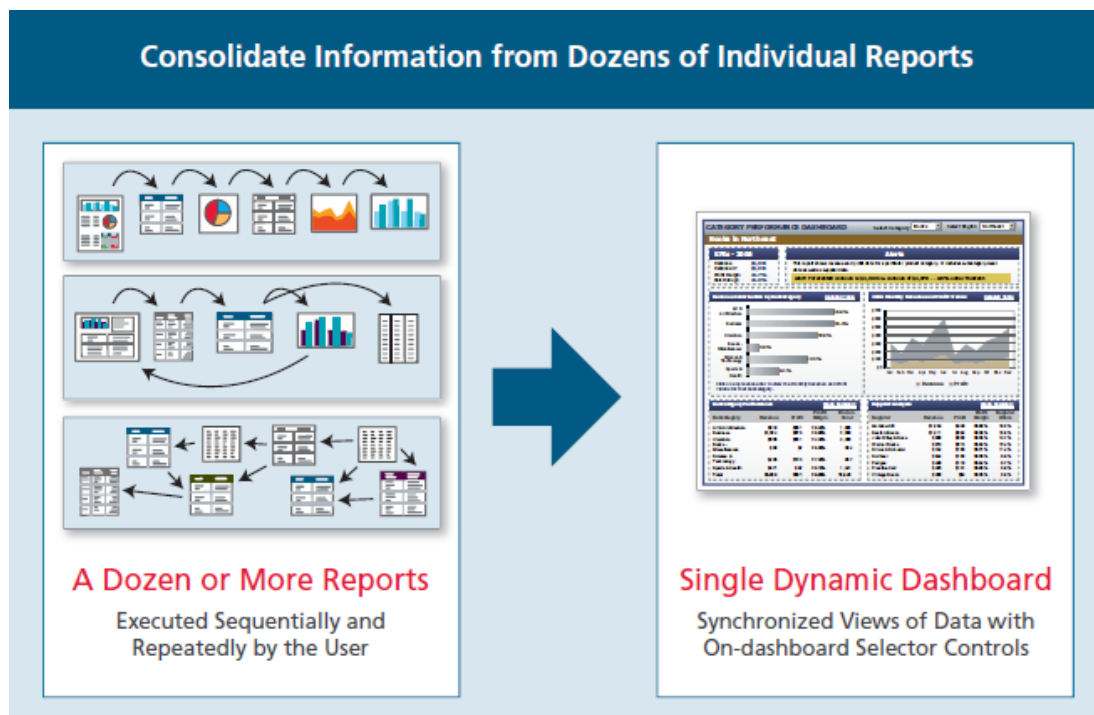
MicroStrategy OLAP Services uses the concept of Intelligent Cube, an in-memory version of a report that can be manipulated by the Analytical Engine. MicroStrategy Developer, MicroStrategy Web, and MicroStrategy Office users can slice and dice data in reports within the Intelligent Cubes without having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



For information on OLAP Services, see the [In-memory Analytics Guide](#).

MicroStrategy Report Services

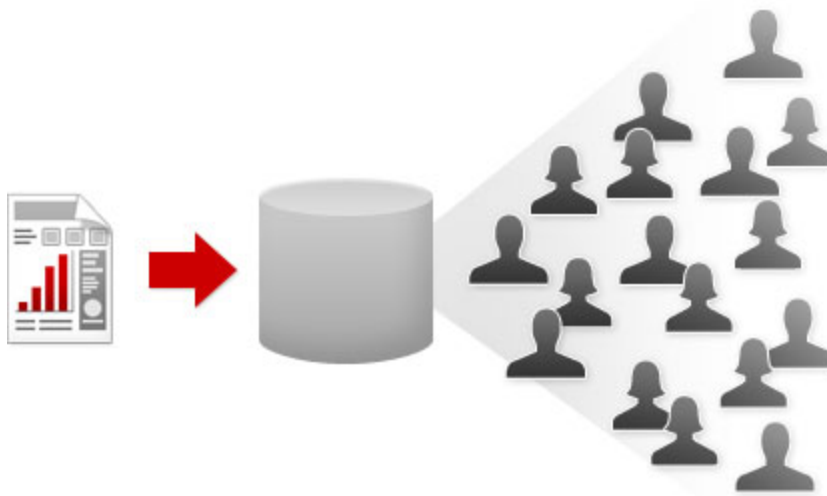
MicroStrategy Report Services is the enterprise reporting engine of the MicroStrategy business intelligence platform. A MicroStrategy Report Services document contains objects representing data coming from one or more reports, as well as positioning and formatting information. It is used to format data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



For information on Report Services, see the [Document Creation Guide](#) and the [Dashboards and Widgets Creation Guide](#).

MicroStrategy Distribution Services

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



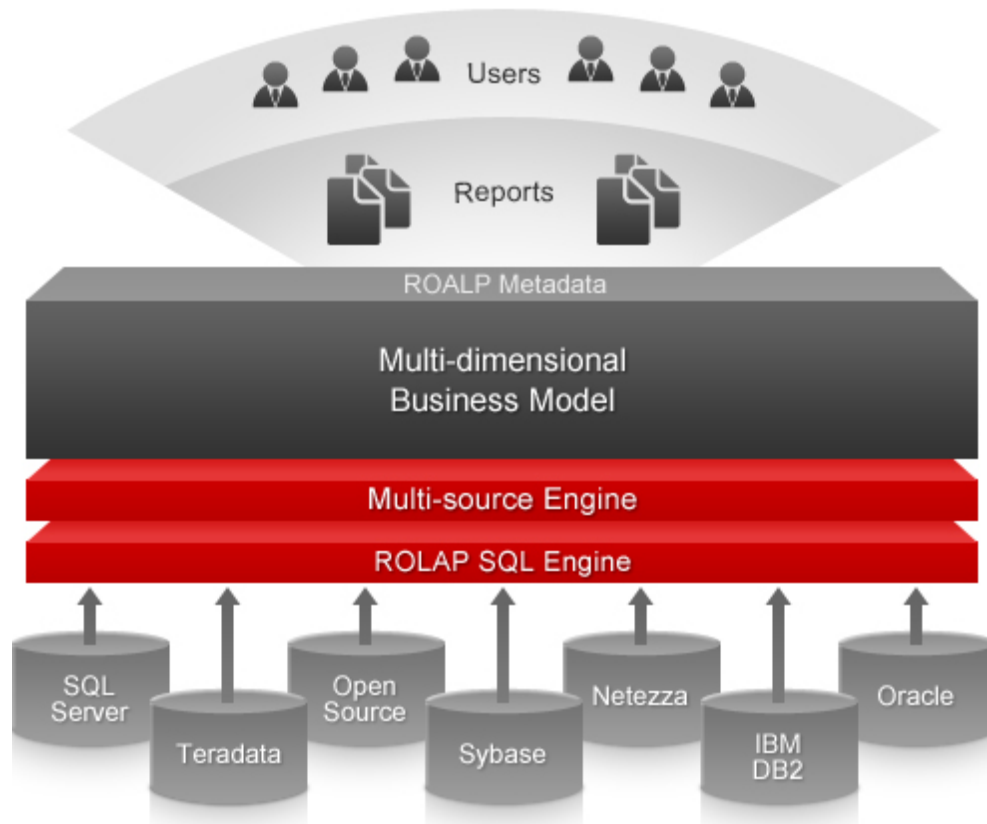
MicroStrategy Transaction Services

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

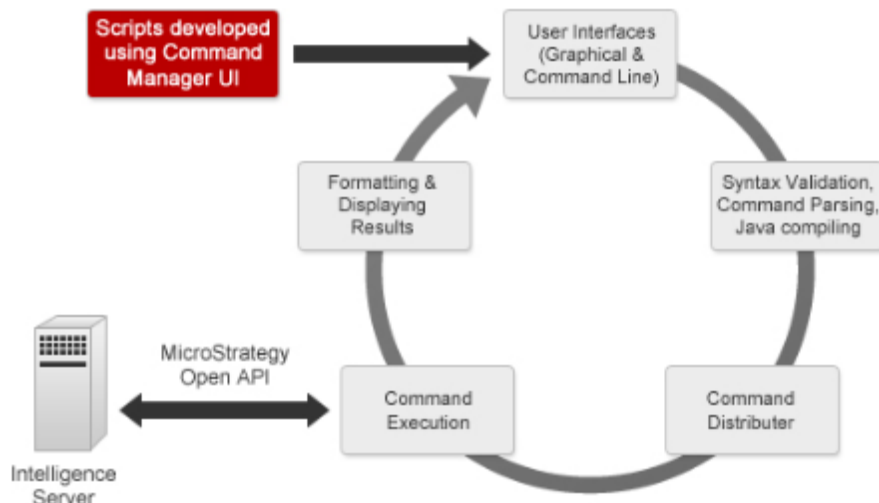
With MultiSource Option, you can connect a project to multiple relational data sources. This allows you to integrate all your information from various databases and other relational data sources into a single MicroStrategy project for reporting and analysis purpose. All data sources included using the 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

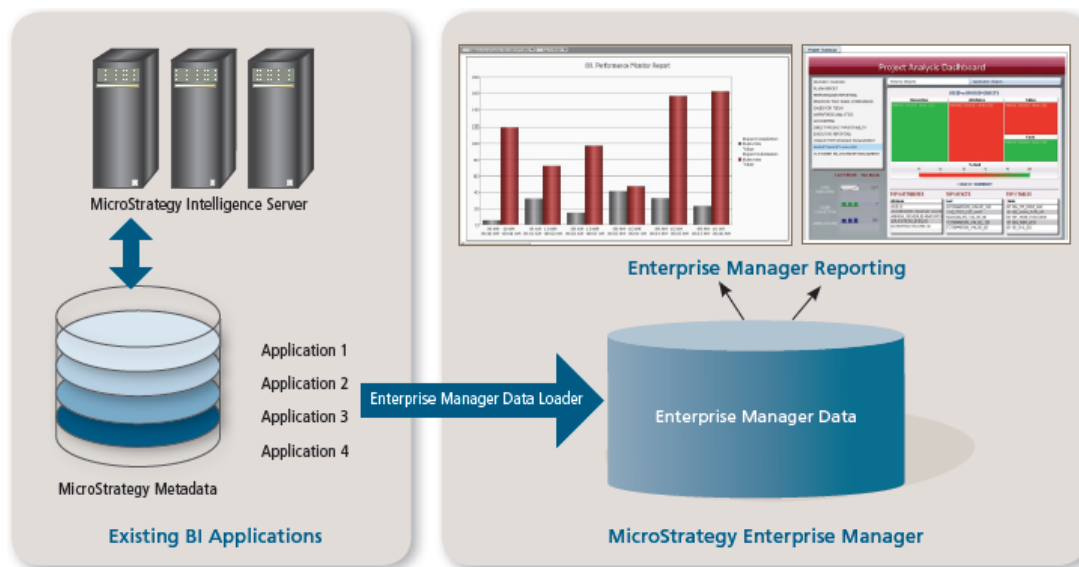
MicroStrategy Command Manager is an application designed to simplify and automate administration tasks, such as add, delete, or update enterprise-level data associated with large numbers of users and user groups. Additionally, Command Manager allows you to manage various configuration settings within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

MicroStrategy System Manager

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

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

For information on using MicroStrategy System Manager to configure and deploy your MicroStrategy environments, see the [System Administration Guide](#).

MicroStrategy Narrowcast Server

MicroStrategy Narrowcast Server proactively distributes personalized information to employees, business partners, and customers through a variety of devices, including mobile phones, email, and Web pages. The distribution of personalized messages and targeted offers is triggered according to predefined schedules and exception criteria, delivering

information in a timely and convenient manner. Narrowcast Server also provides a self-subscription portal, easing administrative responsibilities and empowering information consumers to choose the information they receive. Narrowcast Server can draw information from relational or non-relational sources.

Subscriptions can also be supported through Intelligence Server with the introduction of Distribution Services. For information on Distribution Services, see [MicroStrategy Distribution Services, page 32](#).

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

SequeLink ODBC Socket Server

SequeLink is a complete, end-to-end solution for configuring and managing data access across virtually any number of data stores, operating systems, and deployment options. SequeLink ODBC Socket Server is required to support MicroStrategy Narrowcast Server. It can also be used to access Microsoft Access databases and Microsoft Excel files stored on a Windows machine from an Intelligence Server hosted on a Linux machine (see [MicroStrategy ODBC Driver for SequeLink, page 403](#)).

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 lifecycle, 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 Architect, page 36.*
 - 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> 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.

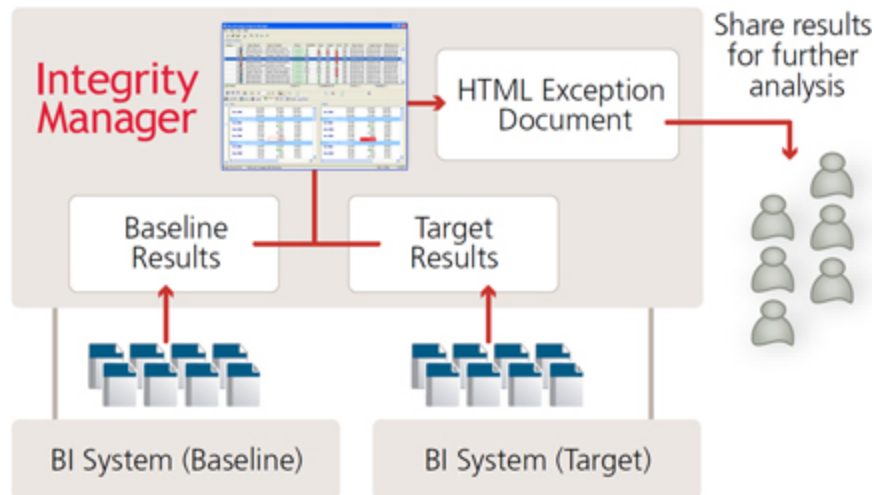
MicroStrategy Architect

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

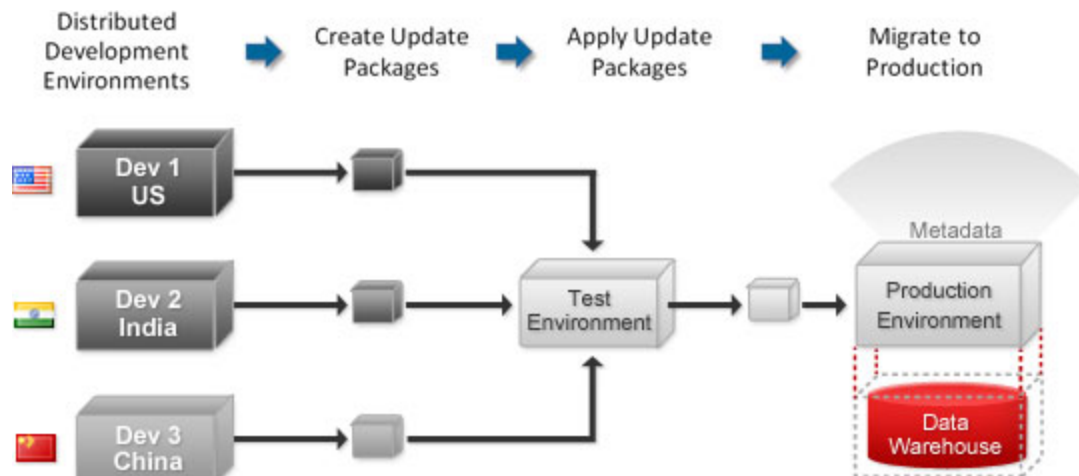
MicroStrategy Integrity Manager is an automated report comparison tool. Report SQL, report data, and graphs can be executed and compared in Integrity Manager to help customers verify change success. In addition, the report comparison output can be analyzed at the report level in MicroStrategy Integrity 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.

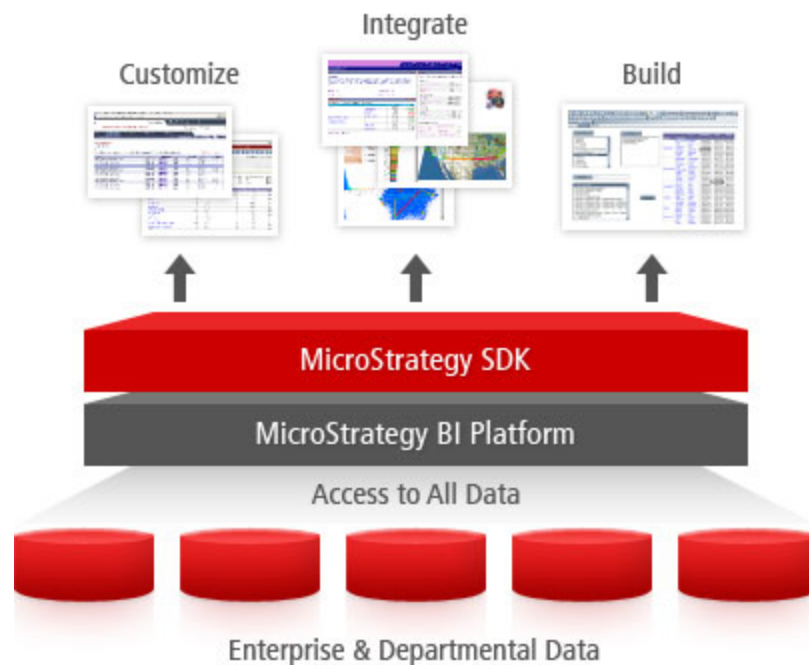


For information on using Object Manager, see the [System Administration Guide](#).

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>. 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.

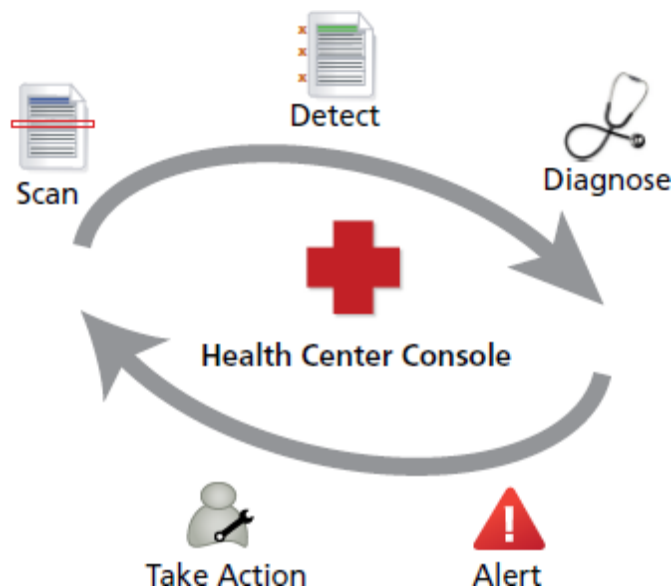
The theme of the Tutorial project is a retail store that sells electronics, books, movies, and music. The key features include:

- 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, see [Installing and Configuring Usher](#).

Messaging Services is a component that is coupled with the Intelligence Server during installations and upgrades. Messaging Services is configured out-of-the-box and runs automatically after the installation is completed. Logs will be sent to MicroStrategy Messaging Services Server only when Messaging Services feature is enabled and Kafka Server can be connected successfully. Logs will be sent to local disk if Messaging Services is disabled or the Kafka Server is down or unreachable because of network issues.

Windows

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

- **Apache Kafka** (C:\Program Files (x86)\MicroStrategy\Messaging Services\Kafka\kafka_2.11-0.10.1.0)
- **Apache ZooKeeper** (C:\Program Files (x86)\MicroStrategy\Messaging Services\Kafka\kafka_2.11-0.10.1.0)
- **MicroStrategy Intelligence Server Log Consumer** (C:\Program Files (x86)\MicroStrategy\Intelligence Server\KafkaConsumer)

Configuring Messaging Services after upgrading

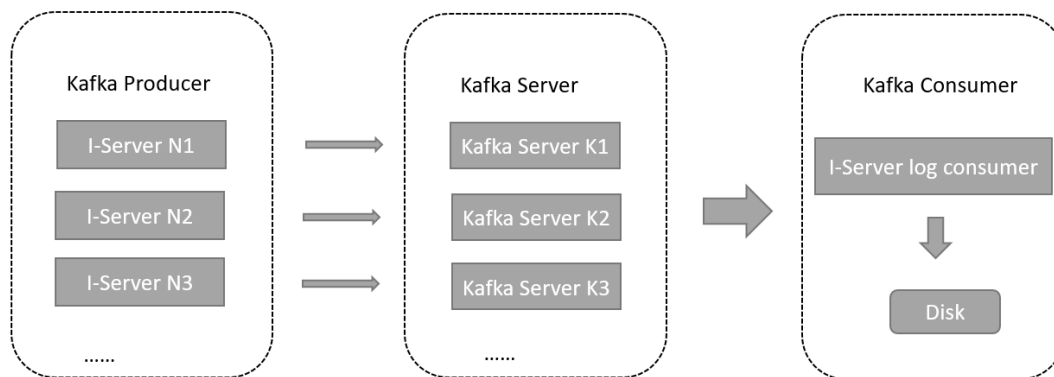


By default, MicroStrategy Messaging Services are installed along with the Intelligence server upgrade.

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

Messaging Services Workflow for Intelligence Server

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



How to Enable Messaging Services

Messaging Services feature configuration is saved in MicroStrategy Intelligence Server configuration, it can be enabled or disabled on the fly, without restarting your Intelligence Server. When the Messaging Services feature is enabled, you will see Kafka log files created in the Kafka installation folder:

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

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

Command Manager Scripts for Messaging Services


To check if Messaging Services is enabled:

```
list all properties for server configuration;
```

To enable Messaging Services through Command Manager script

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

In the example above set:

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


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

Modifying Messaging Services Configuration

Apache Kafka Server

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

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

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

Intelligence Server Log Consumer

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

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

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


```
C:\Program Files (x86)\MicroStrategy\Intelligence
Server\KafkaConsumer>java -jar KafkaConsumer.jar
```
2. Modify file `C:\Program Files (x86)\MicroStrategy\Intelligence
Server\KafkaConsumer\LogConsumer.properties` directly.

The default values after installation are:

```
folder_path=C:\\Program Files (x86)\\Common
Files\\MicroStrategy\\Log\\DSSerrors # indicate log
file location
is_silent_mode=true # indicate run consumer in silent
mode
```

```
broker_port=9092 # Kafka Server port number
broker_hostname=127.0.0.1 # Kafka Server IP
poll_time_out=1000 # consumer connection time out
limit in seconds
max_file_size_M=20 # max log file size in MB
max_num_bak=1 # number of backup files
```

MicroStrategy Messaging Services Configuration for Clustered Environments

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

The minimum number of nodes for a cluster is 3.

Each node must have the following installed:

- MicroStrategy Messaging Services
- Apache Kafka
- Apache Zookeeper

Configure Zookeeper

1. Browse to folder `C:\Program Files (x86)\MicroStrategy\Messaging Services\Kafka\kafka_2.11-0.10.1.0\config`.
2. Edit file `zookeeper.properties` by adding following lines:

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



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

3. Go to folder `C:\Program Files (x86)\MicroStrategy\MessagingServices\Kafka\kafka_2.11-0.9.0.1\config\zookeeper`.
4. Create a file named `myid` containing the identifying value from the server parameter name in the `zookeeper.properties` file.

Configure Kafka

1. Browse to folder `C:\Program Files (x86)\MicroStrategy\Messaging Services\Kafka\kafka_2.11-0.10.1.0\config`.
2. Edit file `server.properties`, add a row `zookeeper.connect=10.27.20.16:2181,10.27.20.60:2181,10.15.208.236:2181` to the **Zookeeper** section.

```
##### Zookeeper #####
# Zookeeper connection string (see zookeeper docs for
# details).
# This is a comma separated host:port pairs, each
# corresponding to a zk
# server. e.g.
# "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002".
# You can also append an optional chroot string to
# the urls to specify the
# root directory for all kafka znodes.
# zookeeper.connect=localhost:2181
zookeeper.connect
=10.27.20.16:2181,10.27.20.60:2181,10.15.208.236:2181
```

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

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

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

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

Linux

Messaging Services is a component that is coupled with the Intelligence Server during installations and upgrades. Messaging Services is configured out-of-the-box and runs automatically after the installation is completed. Logs will be sent to MicroStrategy Messaging Services Server only when Messaging Services feature is enabled and Kafka Server can be connected successfully. Logs will be sent to local disk if Messaging Services is disabled or the Kafka Server is down or unreachable because of network issues.

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

- **Apache Kafka**
(
/opt/mstr/MicroStrategy/install/MessagingServices/Kafka/kafka_
2.11-0.10.1.0)
- **Apache ZooKeeper**
(
/opt/mstr/MicroStrategy/install/MessagingServices/Kafka/kafka_
2.11-0.10.1.0)
- **MicroStrategy Intelligence Server Log Consumer** (
/opt/mstr/MicroStrategy/install/IntelligenceServer/KafkaConsum
er)

Configuring Messaging Services after upgrading

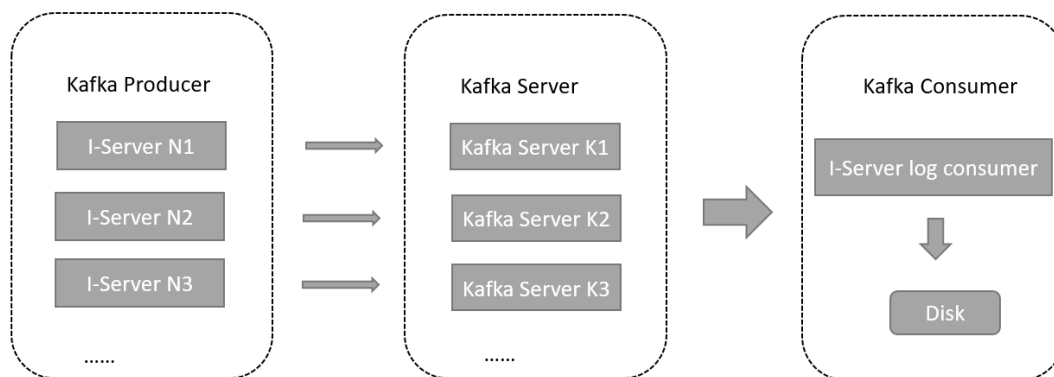


By default, MicroStrategy Messaging Services are installed along with the Intelligence server upgrade.

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

Messaging Services Workflow for Intelligence Server

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



How to Enable Messaging Services

Messaging Services feature configuration is saved in MicroStrategy Intelligence Server configuration, it can be enabled or disabled on the fly, without restarting your Intelligence Server. When the Messaging Services feature is enabled, you will see Kafka log files created in the Kafka installation folder:

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

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

Command Manager Scripts for Messaging Services

To check if Messaging Services is enabled:

```
list all properties for server configuration;
```

To enable Messaging Services through Command Manager script

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

In the example above set:

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



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

Modifying Messaging Services Configuration

Apache Kafka Server

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

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



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

Intelligence Server Log Consumer

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

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

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

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

2. Modify file `/opt/mstr/MicroStrategy/install/IntelligenceServer/KafkaConsumer/LogConsumer.properties` directly.

The default values after installation are:

```
max_num_bak=1 #indicate the number of back up files
max_file_size_M=20 #indicate the maximum file size in MB
broker_port=9092 #Kafka Server port number
is_silent_mode=true #indicate run consumer in silent mode
folder_path=/opt/mstr/MicroStrategy/log/DSSErrors #indicate log folder location
broker_hostname=127.0.0.1 #Kafka Server IP
poll_time_out=1000 #consumer connection time out limit in seconds
```

MicroStrategy Messaging Services Configuration for Clustered Environments

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

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

Each node must have the following installed:

- MicroStrategy Messaging Services
- Apache Kafka
- Apache Zookeeper

Configure Zookeeper

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

```
maxClientCnxns=0
```



```

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

```

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

3. Go to folder
`/opt/mstr/MicroStrategy/install/MicroStrategy/MessagingServices/Kafka/kafka_2.11-0.9.0.1/tmp/zookeeper.`
4. Create a file named `myid` containing the identifying value from the server parameter name in the `zookeeper.properties` file.

Configure Kafka

1. Browse to folder
`/opt/mstr/MicroStrategy/install/MicroStrategy/MessagingServices/Kafka/kafka_2.11-0.9.0.1/config.`
2. Edit file `server.properties`, add a row
`zookeeper.connect=10.27.20.16:2181,10.27.20.60:2181,10.15.208.236:2181` to the **Zookeeper** section.

```

##### Zookeeper #####
# Zookeeper connection string (see zookeeper docs for
# details).
# This is a comma separated host:port pairs, each
# corresponding to a zk
# server. e.g.
# "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002".
# You can also append an optional chroot string to
# the urls to specify the
# root directory for all kafka znodes.
# zookeeper.connect=localhost:2181
zookeeper.connect
=10.27.20.16:2181,10.27.20.60:2181,10.15.208.236:2181

```

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

```

##### Server Basics #####
# The id of the broker. This must be set to a unique
# integer for each broker.
broker.id=2

```

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

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

To execute a service command for Kafka Server and Zookeeper, enter:
`/etc/init.d/kafka-zookeeper {start|stop|status}.`

To execute a service command for MicroStrategy Intelligence Server Log Consumer, enter:
`/etc/init.d/consumer-iserver {stop/start/status}.`

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 77](#).
- Installation location for your MicroStrategy products

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

- **Windows:**
 - You must log on to your machine using a domain account with Windows administrative privileges for the domain or target machine.
 - The user installing MicroStrategy needs write permissions in the installation directory to complete the installation; otherwise the installation fails.
- **Linux:**
 - You need root access permissions for installation if you have purchased the CPU-based MicroStrategy license.
 - You need root access permissions to install Usher components.

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

- [Recommended installation location and example deployments, page 51](#)
- [Hardware requirements and recommendations, page 54](#)
- [Software requirements and recommendations, page 60](#)

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:

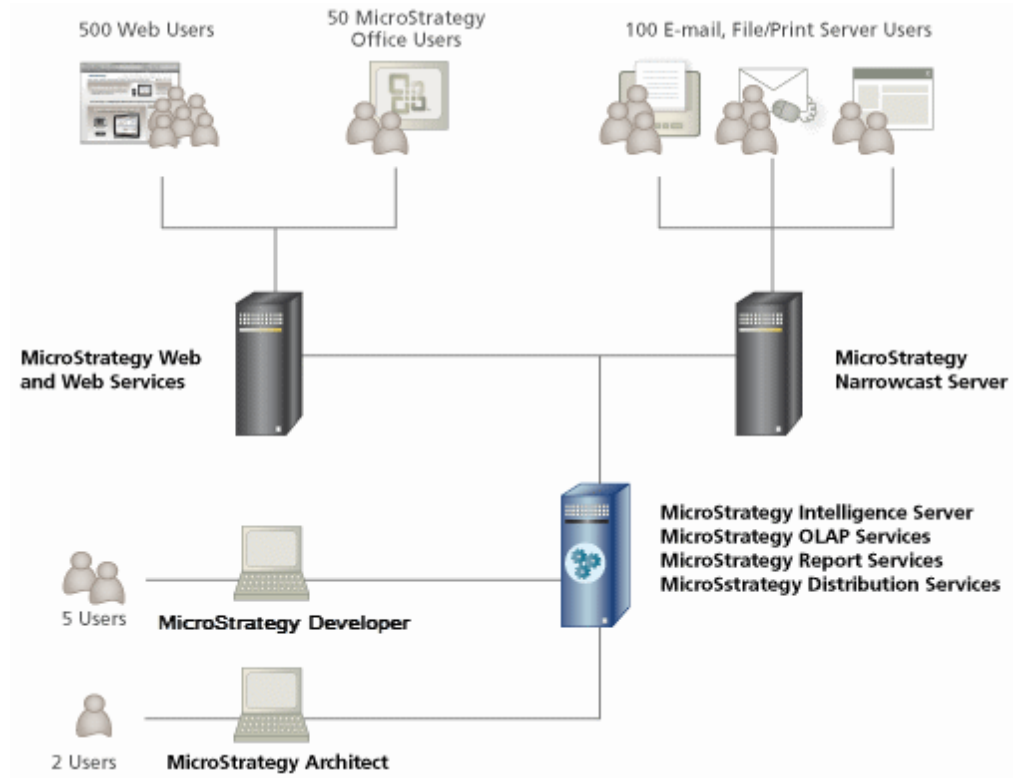
- Intelligence Server should be installed on its own dedicated server machine.
- MicroStrategy Web should be installed on its own dedicated Web server machine.
- The rest of the products can be installed in varying combinations depending on who intends to use them and on what machines.



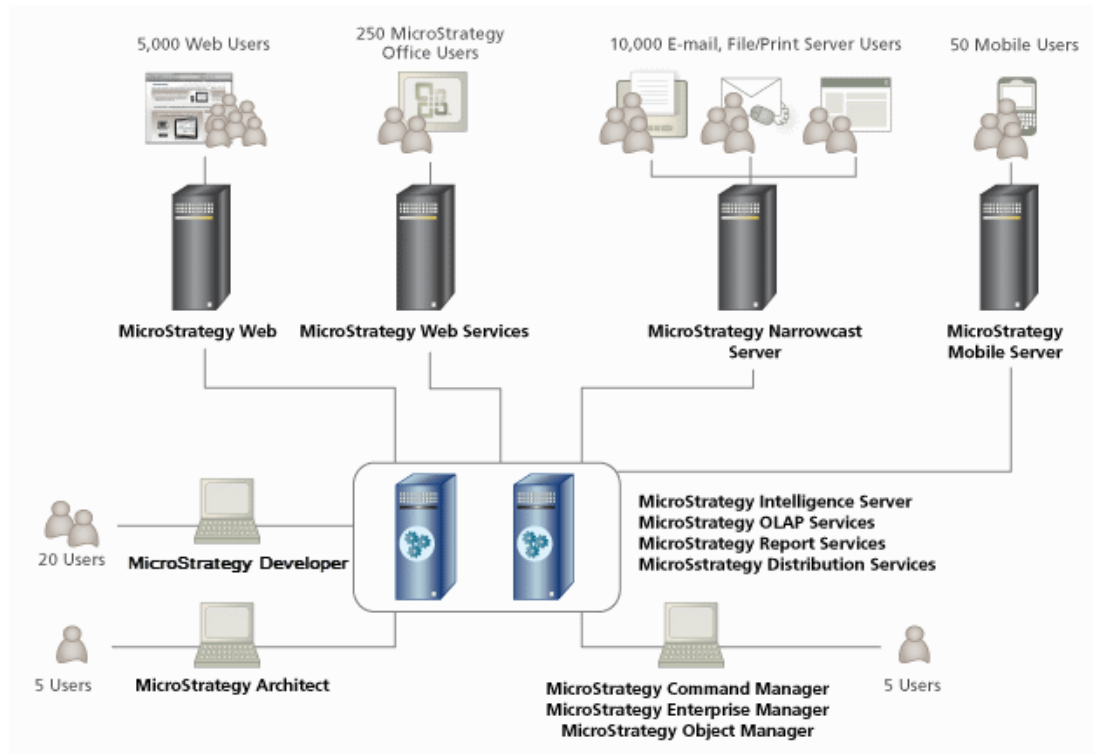
These are just suggestions to help you get started. Read the rest of this chapter for more detailed guidelines.

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.

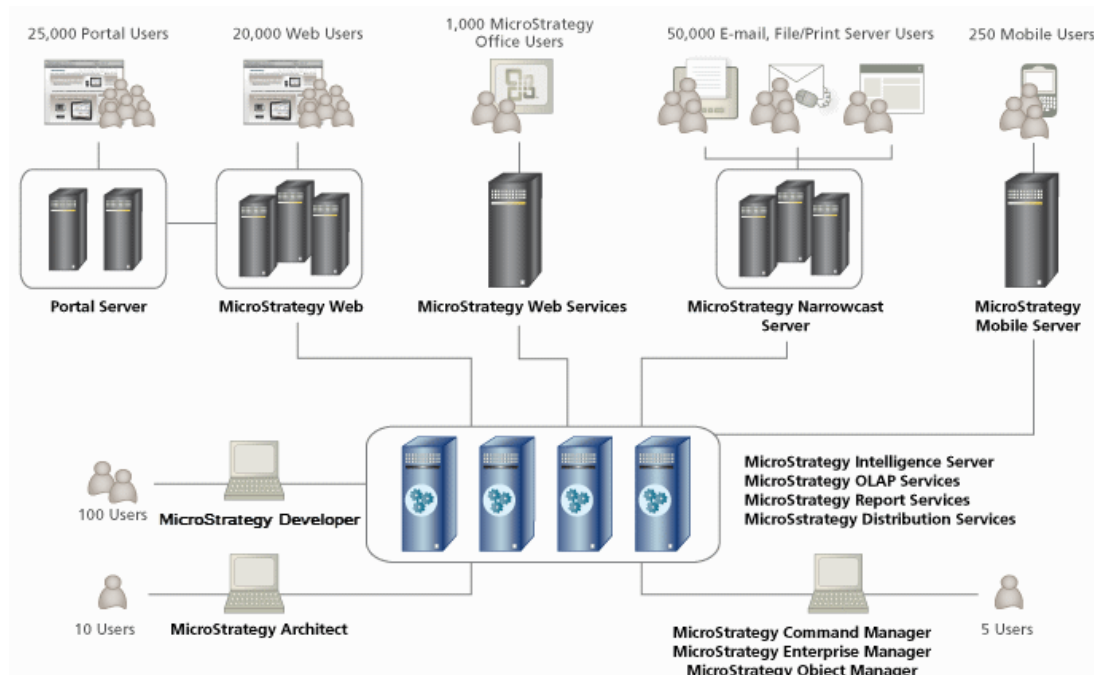
Small production deployment



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 Linux systems, several system settings can affect the performance of MicroStrategy Intelligence Server. These settings do not need to be set before a MicroStrategy installation. For more information on these settings and their recommended values, see [Recommended system settings for Linux, page 74](#).

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.

MicroStrategy Product	Processor	Memory	Storage
MicroStrategy System Manager	x86 or x64 compatible	2 GB	0.5 GB
MicroStrategy Command Manager	x86 or x64 compatible	2 GB	0.25 GB
MicroStrategy Enterprise Manager	x64 compatible	1 GB	0.25 GB
MicroStrategy Object Manager	x86 or x64 compatible	1 GB	0.25 GB
MicroStrategy Developer products	x86 or x64 compatible	2 GB or higher	0.25 GB

MicroStrategy Product	Processor	Memory	Storage
MicroStrategy Intelligence Server	x64 compatible	<p>4 GB or higher</p> <p>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.</p>	<p>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.</p>

MicroStrategy Product	Processor	Memory	Storage
MicroStrategy Integrity Manager	x64 compatible	2 GB or higher	0.25 GB
MicroStrategy Office	x86 or x64 compatible	2 GB	0.5 GB
MicroStrategy Mobile Server	The MicroStrategy Mobile Server hardware requirements are the same as those for MicroStrategy Web Server hardware requirements.		
MicroStrategy Narrowcast Server	x86 or x64 compatible	4 GB	4 GB
MicroStrategy SDK	The SDK is not included in the MicroStrategy installation and can instead be downloaded from the MicroStrategy support site.		
MicroStrategy Analytics Modules	Not applicable	Not applicable	0.5 GB
MicroStrategy Web: Web Client	x86 or x64 compatible	2 GB or higher	MicroStrategy Web can be accessed from a third-party web browser, which means there are no additional storage requirements.
MicroStrategy Web: Web Server	x64 compatible	4 GB or higher	0.5 GB

System hardware requirements and recommendations on Linux

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



- The storage recommendations listed in the table below provide an estimate for installing and supporting each MicroStrategy product on a separate machine. For information on the total size of a MicroStrategy installation when installing all MicroStrategy products on the same machine, see below.
- A successful configuration of Intelligence Server depends on a valid combination of an operating system and a CPU architecture. Valid operating system and CPU architecture combinations for Intelligence Server are listed in the table below.
- The storage requirements listed in the table below for Intelligence Server include additional space than is required for the initial installation. This additional space is to support the creation of the various files that these products require throughout their use in a MicroStrategy environment. Additional space may be required depending on the use of Intelligence Server in your MicroStrategy environment.

MicroStrategy Product	Processor	Memory	Storage Recommendation
MicroStrategy System Manager	Linux: x86-64 compatible	2 GB or higher	3GB on other Linux operating systems
MicroStrategy Command Manager	Linux: x86-64 compatible	2 GB or higher	3GB on other Linux operating systems
MicroStrategy Intelligence Server	Linux: x86-64 compatible	4 GB or higher 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	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.

MicroStrategy Product	Processor	Memory	Storage Recommendation
		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.	
MicroStrategy Integrity Manager	Linux: x86-64 compatible	2 GB or higher	3 GB on other Linux operating systems
MicroStrategy Web Services for Office	Linux: x86-64 compatible	2 GB or higher	3 GB on other Linux operating systems
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	Linux: x86-64 compatible	4 GB or higher	3 GB on other Linux operating systems
MicroStrategy Web: Web Client	Linux: x86-64 compatible	2 GB or higher	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 Linux

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

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

Temporary directory requirements for installation

In addition to the space requirements listed above, you also need free space in the temporary directory. The default location of the temporary directory is `/tmp`.

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

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



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

MicroStrategy Mobile hardware requirements for mobile devices

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

Flash memory

Requirement	Minimum	Recommended
Flash memory	32 MB	64 MB

Android devices

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

Software requirements and recommendations

See the MicroStrategy [Readme](#) for the specific software requirements and recommendations for MicroStrategy products on the Windows, UNIX, and Linux platforms.

Intelligence Server software requirements on Linux

For the exact information such as version numbers and space requirements, see the MicroStrategy [Readme](#).

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

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

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

The following requirements also apply to all Linux platforms:

- A Web browser (for example, Firefox) is required for viewing the MicroStrategy Readmes and online help.
- Windows Services for UNIX (<http://technet.microsoft.com/en-us/interopmigration/bb380242.aspx>) or Samba (<http://www.samba.org>) is required for HTML document support. Samba 3.0 is required for the support of HTML documents with alphanumeric names.

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

- [Configuring shared memory resources, page 61](#)
- [Linux, page 63](#)

Be aware of the following before reviewing the sections listed above:



- The operating systems listed are deemed supported or certified to reflect the level of internal testing that each configuration was exposed to for the current release. MicroStrategy recommends using certified configurations over the supported configurations.
- MicroStrategy certifies and supports operating systems that are compatible with a set of CPU chipsets, referred to as CPU architectures, that are binary-compatible. MicroStrategy tests on at least one of the CPU chipsets within a set of binary-compatible CPU architectures for purposes of certifying and supporting operating systems with MicroStrategy products. A valid CPU architecture is provided in parentheses () to clarify the operating system software certified or supported for Intelligence Server.
- All Linux operating systems are 64-bit.
- For information on LDAP Servers certified and supported for LDAP authentication with various Intelligence Server machine environments, see the *MicroStrategy Readme*.

Configuring shared memory resources

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

During installation (on the System Requirements page, see [Installing with the MicroStrategy Installation Wizard, page 112](#)), you have the following options:


- **Exit the MicroStrategy setup wizard to do the required system changes (Recommended):** Select this option to cancel the installation and make the required system resource limit changes to support shared memory resources. This option is recommended for production environments. Information on the recommended resource limits is below.
- **Allow the setup to reconfigure MicroStrategy to use Pipe as the Default IPC Mechanism:** Select this option to disable the use of shared memory resources for Intelligence Server, and instead use the pipe mechanism. Disabling the ability to use shared memory resources can decrease the performance of your MicroStrategy applications; therefore, this is not recommended for production environments.
- **Keep Shared Memory as the Default IPC Mechanism. (MicroStrategy may not work properly):** Select this option to keep your system resource limits set at their current values to support shared memory resources. While this allows you to continue installation with the current system resource limits, Intelligence Server may not function properly after installation. If you plan to use shared memory resources for enhanced performance of your production environments, you should select the first option to exit the installation and make the required system changes.

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



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

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

Setting Name	Description	Recommended Value
shmmni	Maximum number of shared memory identifiers at any given time.	4096
shmseg	Maximum number of segments per process.	2048
		 This setting does not exist on 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:

Setting Name on Linux	Description	Recommended Value
semmsl	Maximum number of semaphores in a semaphore set.	250

Setting Name on Linux	Description	Recommended Value
semmns	Maximum number of semaphores in the system.	32000
semopm	Maximum number of operations in a simple semaphore call.	32
semmni	Maximum number of semaphore sets.	4096

Linux

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

Supporting Intelligence Server memory allocation on Linux

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

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

MicroStrategy Web JSP software requirements and recommendations

To confirm the latest requirement information, see the *MicroStrategy Readme*. For specific patches, filesets, technology level, and other requirements for UNIX and Linux operating systems, see [Intelligence Server software requirements on Linux, page 60](#).

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 63](#)).

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 Http 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 a noticeable reduction in system performance.

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

- Ratio of MicroStrategy Web servers to Intelligence Servers—In a typical system you should have a 1:1 ratio of Intelligence Servers to MicroStrategy Web servers. This ensures that resources on both sides are optimized. However, you might find it useful to add Intelligence Servers or MicroStrategy Web servers depending on your particular requirements.
- Report Styles—MicroStrategy Web provides a set of different XSL report styles. These styles provide an easy way for you to customize how reports look. Due to the varying complexity of these styles, some might require more processing than others.

MicroStrategy Professional Services for high performance

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

- Employ in-memory Business Intelligence
- Design high performance dashboards
- Optimize query efficiency
- Implement effective caching strategies
- Configure MicroStrategy for high performance

In just one week, MicroStrategy Professional Services, will conduct a thorough examination of your Business Intelligence implementation, providing you with actionable recommendations on these five key areas to improve overall performance. The MicroStrategy Performance Analysis service delivers:

- **Performance optimization roadmap:** A customized report with prioritized recommendations to achieve performance goals.
- **System configuration:** Optimum configuration setting recommendations to achieve efficient use of resources across different MicroStrategy products.
- **Performance monitoring plan:** A set of performance related metrics to proactively monitor and identify performance opportunities.

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

Common questions about sizing

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

Why should I increase the processor speed of Intelligence Server?

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

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

Why should I add more processors to Intelligence Server?

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

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

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

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

For more detailed information on tuning your machine's memory for your MicroStrategy environment, see the section *Managing system resources* in the [System Administration Guide](#).

What would more network bandwidth do for me?

You can tell whether your network is negatively impacting your system's performance by monitoring how much of your network's capacity is being used. To monitor this on a Windows machine, use the Windows Performance Monitor for the object **Network Interface**, and 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.

Installation	Result
Fresh installation on a system in which MicroStrategy application has never been installed before	<p>The MicroStrategy Installation Wizard prompts you to select the language from the drop-down list.</p> <p>The user language in the product interface is the language that you select during installation.</p>
Repair or maintenance installation on a system on which MicroStrategy application has been installed before	<p>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.</p> <p>The user language in the product interface is also the language that you selected the first time you installed the product on the system.</p>
Completely uninstalling all the MicroStrategy products and installing the same version or a newer version	<p>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.</p> <p>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.</p>

 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 10.8 can only connect to Intelligence Server 10.8 or later. For a complete list of compatible MicroStrategy Web and Intelligence Server versions, refer to the *MicroStrategy Readme*.

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

Certified ODBC drivers for MicroStrategy Intelligence Server

MicroStrategy certifies ODBC drivers for Windows and Linux for Intelligence Server and different DBMS types. MicroStrategy-branded ODBC drivers are installed with the MicroStrategy products.

- For a complete list of certified and supported configurations with exact version numbers, refer to the certified and supported configurations listed in the *MicroStrategy Readme*.

Certificates used during Usher Installation and Configuration

Name	Example file path	Where used	Purpose
Usher Server SSL/HTTPS Certificate	/certs/HTTPSServerCertificate.crt	<p>(1) Within the MicroStrategy Installation Wizard, on the page for Usher, the field labeled "SSL Server Certificate" or "SSL Certificate file".</p> <p>(2) Within the Usher Configuration web interface, on the first page, the field labeled "SAML Certificate".</p>	<p>(1) SSL certificate used by the Tomcat web services.</p> <p>(2) SAML certificate provided by Usher Network Manager.</p>

Name	Example file path	Where used	Purpose
Usher Server SSL/HTTPS Certificate Private Key	/certs/HTTPSServerCertificate.key	<p>(1) Within the MicroStrategy Installation Wizard, on the page for Usher, the field labeled "SSL Server Certificate Key" or "Private Key File".</p> <p>(2) Within the Usher Configuration web interface, on the first page, the field labeled "SAML Key".</p>	<p>(1) SSL private key for the certificate used by the Tomcat web services.</p> <p>(2) SSL private key for the SAML certificate provided by Usher Network Manager.</p>
Third-Party Certificate Authority Root Certificate	/certs/ca-root.crt	Prior to running the MicroStrategy Installation Wizard, include it in the Usher Server Certificate Authority Chain (see below).	SSL certificate representing the Certificate Authority that signed the Usher Server SSL/HTTPS Certificate. Each CA has 1 root certificate.
Third-Party Certificate Authority Intermediate Certificate	/certs/ca-intermediate.crt	Prior to running the MicroStrategy Installation Wizard, include it in the Usher Server Certificate Authority Chain (see below).	SSL certificate representing the Certificate Authority that signed the Usher Server SSL/HTTPS Certificate. Each CA may have 1 or more intermediate certificates.
Usher Server Signing Certificate Authority Certificate	/certs/ushersigningca.crt	Within the Usher Configuration web interface, on the first page, the field labeled "SSL Certificate Authority Certificate".	SSL certificate used by the Usher Security Server to sign any Certificate Signing Request (Usher Security mobile app, Directory Agent, Usher SDK-based apps, VPN Agent, etc.)

Name	Example file path	Where used	Purpose
Usher Server Signing Certificate Authority Private Key	/certs/ushersigningca.key	Within the Usher Configuration web interface, on the first page, the field labeled "SSL Certificate Authority Key".	SSL private key for the Usher Server Signing CA.
Usher Server Certificate Authority Chain	/certs/UsherCAChain.pem	Within the MicroStrategy Installation Wizard, on the page for Usher, the field labeled "CA Certificate Chain" or "SSL Certificate Chain file".	Certificate store that includes all of the Certificate Authority certificates that the Tomcat web services will trust. Includes the third-party CA and Usher Server Signing CA.
Note: This table does not include files used for Usher components deployed outside of the Usher Security Server, such as the Directory Agent, Physical Access Adapters, and third-party Web/Mobile applications implementing Usher APIs.			

Recommended system settings for Linux


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

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

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



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

Setting Name ulimit name (limit name)	Description	Recommended Value
cputime (time)	Maximum CPU seconds per process	Unlimited
filesize (file)	Maximum size for a single file	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).
datasize (data)	Maximum heap size per process	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).
stacksize (stack)	Maximum stack size per process	200 MB
coredumpsize (coredump)	Maximum size for a single core dump file	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.
memoryuse (memory)	Maximum size of physical memory allotted per process	Unlimited, or as large as the physical memory of your system allows
vmemoryuse (vmemory)	Maximum size of virtual memory allowed per process	Unlimited, or as large as your system virtual memory allows
descriptors (nofiles)	Maximum number of file descriptors (open files) per process	8192
processes (per user)	Maximum number of processes per user	8194  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 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 324 in Chapter 9, Automated Installation on Windows](#) and [Silent installation, page 330 in Chapter 10, Automated Installation on Linux](#).

Creating custom installation packages

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

The steps below show you how to create these custom installation packages.



If you are performing a MicroStrategy 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 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 78](#).
 - For Linux installations, the table below lists which installation files are required for each MicroStrategy component. Once you determine the required installation files, you can include them in your custom installation as described in [To specify the location of the installation files, page 78](#) below.

Installation File	MicroStrategy Components That Require The Installation File
mstr1.tzp	All MicroStrategy components and products

Installation File	MicroStrategy Components That Require The Installation File
mstr3.tzp	MicroStrategy Intelligence Server and all of its components
mstr4.tzp	MicroStrategy Web, including Web Analyst, Web Reporter, and Web Professional
mstr5.tzp	MicroStrategy Web Services for Office
mstr6.tzp	MicroStrategy Command Manager
mstr7.tzp	MicroStrategy Integrity Manager
mstr8.tzp	MicroStrategy System Manager
mstr9.tzp	MicroStrategy Mobile Client
mstr10.tzp	MicroStrategy Mobile Server
mstr11.tzp	MicroStrategy Portlets, which is a component of MicroStrategy Web
mstr12.tzp	MicroStrategy GIS Connectors, which is a component of MicroStrategy Web
mstr13.tzp	All MicroStrategy components and products

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 296](#).
 - For Linux installations, you can specify the location of these files using an `options.txt` file, as described in [Providing installation files for smaller installations, page 347](#).

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 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.

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

Complete	Chapter and Installation Task
	<p>Chapter 2, Installing MicroStrategy on Windows: This chapter describes the procedures for installing the MicroStrategy products necessary to run your business intelligence application in a Windows environment.</p> <p>Or</p> <p>Chapter 9, Automated Installation on Windows: As an alternative to the regular installation, you can perform a fully automated and unattended installation including customization routines available with the product. This chapter describes different types of unattended and automated installations and provides steps to perform these installations on Windows.</p> <p>Additionally, Chapter 11, Deploying OEM Applications explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.</p>
	<p>Chapter 5, Activating Your Installation: After installing MicroStrategy products, you have 30 days to activate your software installation. If you have not activated your software after 30 days, some MicroStrategy features may become unavailable until you complete the software activation.</p>
	<p>Chapter 6, Configuring and Connecting Intelligence Server: After installing and activating MicroStrategy products, you must use the MicroStrategy Configuration Wizard to configure the MicroStrategy metadata repository, statistics tables, history list tables, Intelligence Server, and project sources. This chapter describes the steps used to configure an installed MicroStrategy suite of products using the MicroStrategy Configuration Wizard.</p>
	<p>Chapter 7, Deploying MicroStrategy Web and Mobile Server: You can deploy your project to your user community using MicroStrategy Web. This chapter provides information on how to deploy and configure MicroStrategy Web on Windows and Linux platforms with various Web and application servers.</p>
	<p>Chapter 13, Adding or Removing MicroStrategy Components: This chapter describes the steps to add and remove MicroStrategy components on Windows, as well as other operating systems. For Windows platforms, refer to the following sections:</p> <ul style="list-style-type: none"> • Adding or removing MicroStrategy components on Windows, page 383. • Re-installing MicroStrategy components on Windows, page 384. • Uninstalling MicroStrategy components on Windows, page 385.

Installing and configuring MicroStrategy on Linux

If you are installing MicroStrategy on 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.

Complete	Chapter and Installation Task
	<p>Chapter 1, Planning Your Installation: Review this chapter for important installation</p>

Complete	Chapter and Installation Task
	prerequisites and considerations.
	<p>Chapter 3, Installing MicroStrategy on Linux: This chapter describes the procedures for installing the MicroStrategy products necessary to run your business intelligence application on a Linux environment.</p> <p>Or</p> <p>Chapter 10, Automated Installation on Linux: As an alternative, you can perform a fully automated and unattended installation without using the graphical user interface. This chapter describes different types of unattended and automated installations and steps to perform these installations on Linux.</p> <p>Or</p> <p>: This chapter describes the full procedures for installing Usher Security in conjunction with other MicroStrategy products.</p> <p>Additionally, Chapter 11, Deploying OEM Applications explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.</p>
	<p>Chapter 5, Activating Your Installation: After installing MicroStrategy products, you have 30 days to activate your software installation. If you have not activated your software after these 30 days have passed, some MicroStrategy features may become unavailable until you complete the software activation.</p>
	<p>Chapter 6, Configuring and Connecting Intelligence Server: After installing and activating MicroStrategy products, you must use the MicroStrategy Configuration Wizard to configure the MicroStrategy metadata repository, statistics tables, history list tables, Intelligence Server, and project sources. This chapter addresses the processes necessary to configure an installed MicroStrategy suite of products using the Configuration Wizard. If no project sources are defined, then the Configuration Wizard opens.</p> <p>Or</p> <p>Chapter 12, Configuring MicroStrategy Using Command Line Tools: MicroStrategy tools are provided in command line mode on Linux so that you can perform various configuration tasks through the operating system console. This enables you to perform your required configurations even if you do not have access to the MicroStrategy interface.</p>
	<p>Chapter 7, Deploying MicroStrategy Web and Mobile Server: You can deploy your project to your user community using MicroStrategy Web. This chapter provides information on how to deploy and configure MicroStrategy Web and Web Universal on Windows and Linux platforms with various Web and application servers.</p>
	<p>Chapter 8, Setting Up Documents and HTML Documents: This chapter explains the setup required for the Intelligence Server to create and execute HTML documents and documents. It also describes the steps to create this setup, which are only necessary on a Linux environment.</p>
	<p>Chapter 13, Adding or Removing MicroStrategy Components: This chapter describes the steps to add and remove MicroStrategy components on all supported operating systems. For Linux platforms, refer to the following section:</p> <ul style="list-style-type: none"> • Uninstalling MicroStrategy components on Linux, page 387

INSTALLING MICROSTRATEGY ON WINDOWS

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, see [Chapter 1, Planning Your Installation](#) for important pre-installation information.

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

- **Windows only:** The Windows only versions, labeled as MicroStrategy Intelligence Server, MicroStrategy Web (ASP.NET), and so on, are compatible only with a Windows platform. With these versions, MicroStrategy Web can be deployed quickly and easily using MicroStrategy's Internet Information Services (IIS) Web Server. The drawback is that IIS is the only Web server that can be used to deploy the Windows only version of MicroStrategy Web.
- **Universal (platform independent):** The universal versions, labeled as MicroStrategy Intelligence Server, MicroStrategy Web (JSP), and so on, are compatible with Windows as well as Linux platforms. Installing the universal versions on Windows lets you deploy MicroStrategy Web with different application and Web server combinations. For example, instead of using IIS to deploy MicroStrategy Web, you can use Apache Tomcat, Oracle 10g, and so on.



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

This chapter has the following sections:

- [Installation procedure in Windows, page 83](#)
- [Configuring your MicroStrategy installation, page 107](#)

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

Installation procedure in Windows

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

- [Installing with the MicroStrategy Installation Wizard, page 83](#)
- [Installation verification, page 98](#)

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

- Prerequisites, see [Installation prerequisites, page 50](#) in [Chapter 1, Planning Your Installation](#).
- Advanced installation functionality, such as installing in an SMS environment or using installation response files, see [Chapter 9, Automated Installation on Windows](#).
- Installing and deploying MicroStrategy Web with other Web and application servers, see [Chapter 7, Deploying MicroStrategy Web and Mobile Server](#).
- Deploying MicroStrategy Web Services ASP.NET and J2EE, see the [MicroStrategy Office User Guide](#).
- Prerequisites and procedures for Usher Security, see [Usher Pre-Installation Instructions](#).
 - If you have not uninstalled previous versions of MicroStrategy products, you are prompted to overwrite them. Click **Yes** to ensure that all products are installed properly. To retain the existing Tutorial metadata repository and warehouse, rename it or move it to another location before you start the installation process.
 - Although MicroStrategy supports Windows Terminal Services, using Windows Terminal Services is not recommended for installation. It can affect the functionality of some MicroStrategy components.

Installing with the MicroStrategy Installation Wizard

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 50](#):

- **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.
- 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 99](#).
 - 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 77](#). Details on using a `response.ini` file to provide the location of the installation files are provided in [Chapter 9, Automated Installation on Windows](#) and the parameters used to specify the location of the required installation files are described in [Installation Files, page 296](#).

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

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

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

Welcome

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 13, Adding or Removing MicroStrategy Components](#).

License Agreement

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



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**



Contact Technical Support to obtain a license key.

Install Options

Select one of the following install options:

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

For steps to use the Express installation, see [Performing a MicroStrategy Express installation](#).

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

Choose Destination Location

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

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

While this setting determines the default root directory for the MicroStrategy products you install, you can change the destination of an individual product later as part of selecting which MicroStrategy products to install.

You can choose the directory for a product only if that product is not already installed on the server machine. Otherwise, the product can only be installed in the same directory in which it already exists.

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

Select Features

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




After you have selected all required MicroStrategy products and defined the proper installation locations, if 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 51 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 30](#))
 - MicroStrategy Intelligence Server subcomponents such as MicroStrategy OLAP Services, MicroStrategy Report Services, MicroStrategy Distribution Services, MicroStrategy Transaction Services, MicroStrategy MultiSource Option, and MicroStrategy Clustering Option.
 - R and R Integration Pack libraries are optional open source components that can be installed during MicroStrategy platform installation. R is used by Intelligence Server to process 'R' based functions to enable 'R' analytics.
- MicroStrategy Web (see [MicroStrategy Web components, page 26](#))
 - MicroStrategy Portlets (see [MicroStrategy Portlets, page 26](#))
 - MicroStrategy GIS Connectors (see [MicroStrategy GIS Connectors, page 27](#))
- MicroStrategy Office (see [MicroStrategy Office, page 27](#))
 - MicroStrategy Web Services for Office (see [MicroStrategy Web Services \(ASP.NET\) and Web Services \(J2EE\), page 28](#))

 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 269 of Chapter 7, Deploying MicroStrategy Web and Mobile Server](#).

- MicroStrategy Mobile (see [MicroStrategy Mobile, page 28](#))
- MicroStrategy Developer Products (see [MicroStrategy Developer, page 35](#))
- MicroStrategy Object Manager (see [MicroStrategy Object Manager, page 37](#))
- MicroStrategy Command Manager (see [MicroStrategy Command Manager, page 33](#))
- MicroStrategy Enterprise Manager (see [MicroStrategy Enterprise Manager, page 34](#))
- MicroStrategy Integrity Manager (see [MicroStrategy Integrity Manager, page 36](#))
- MicroStrategy System Manager (see [MicroStrategy System Manager, page 34](#))
- MicroStrategy Narrowcast Server (see [MicroStrategy Narrowcast Server, page 34](#))
- MicroStrategy Analytics Modules (see [MicroStrategy sample projects, page 39](#))
- Usher Security (see [Usher Help](#))

 This also includes MicroStrategy Tutorial Reporting (see [MicroStrategy Tutorial Reporting, page 39](#))

- 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 Linux machine (see [MicroStrategy ODBC Driver for SequeLink, page 403](#)).



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 19](#) in [Chapter 1, Planning Your Installation](#).

Defining the installation location for MicroStrategy products and sub-components

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 86](#)).

- Provide the location of the installation files using a `response.ini` file. This lets you access the installation files stored on a folder or stored at a URL and accessed using HTTP or HTTPS. Details on using a `response.ini` file as part of an installation are provided in [Chapter 9, Automated Installation on Windows](#) and the parameters used to specify the location of the required installation files are described in [Installation Files, page 296](#).

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.



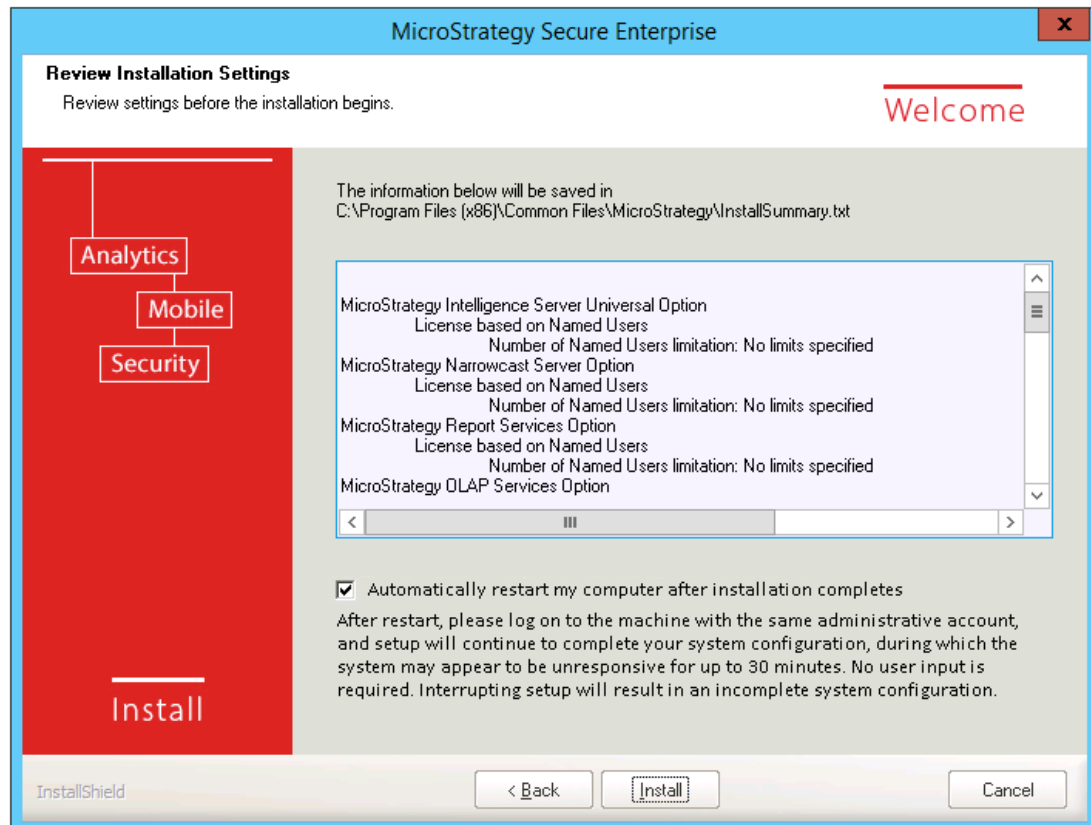
- `openssl.exe x509 -noout -modulus -in sample.crt | openssl.exe md5`
- `openssl.exe rsa -noout -modulus -in sample.key | openssl.exe md5`

- 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`.

Review Installation Settings

This screen provides a summary of the MicroStrategy products and services you have selected to install along with the destination folder where they will be saved. If you wish to automatically restart your computer when the installation is complete, select the check box below the list of products and services as shown below. Restarting your computer after installation will ensure that the system configuration is completed properly.



If you do not choose to restart automatically, you can restart manually when the Installation Wizard is complete.

MicroStrategy 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

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 online help.



- 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 online help.



- 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 148 in Chapter 5, Activating Your Installation](#).

You have a grace period of 30 calendar days to activate your installation. If you do not complete the activation before the grace period expires, your MicroStrategy 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 150 in Chapter 5, 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.

- If you have a previous version of MicroStrategy Web installed on the machine, the new version you install uses the same virtual directory the previous version is using. Therefore, you are not prompted to specify the name of the virtual directory.
- The name provided for a virtual directory must be unique. You cannot use the same name as the default for other MicroStrategy products.
- MicroStrategy automatically configures the MicroStrategy Web virtual directory to run with the version of .NET Framework that it requires.

MicroStrategy Web (ASP.NET) CPU Affinity Setting

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

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

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


For more information on the MicroStrategy Web CPU affinity feature, refer to the [System Administration Guide](#).

MicroStrategy Mobile Server (ASP.NET) Setting

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

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


Mobile Server can be deployed using the same techniques used to deploy MicroStrategy Web, as described in [Deploying MicroStrategy Web and Mobile Server, page 214](#). For additional configurations required to deploy Mobile Server, see the [MicroStrategy Mobile Design and Administration Guide](#).


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

MicroStrategy Subscription Portal Setting

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

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


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

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

MicroStrategy 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 95](#).

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.aspx
```

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.

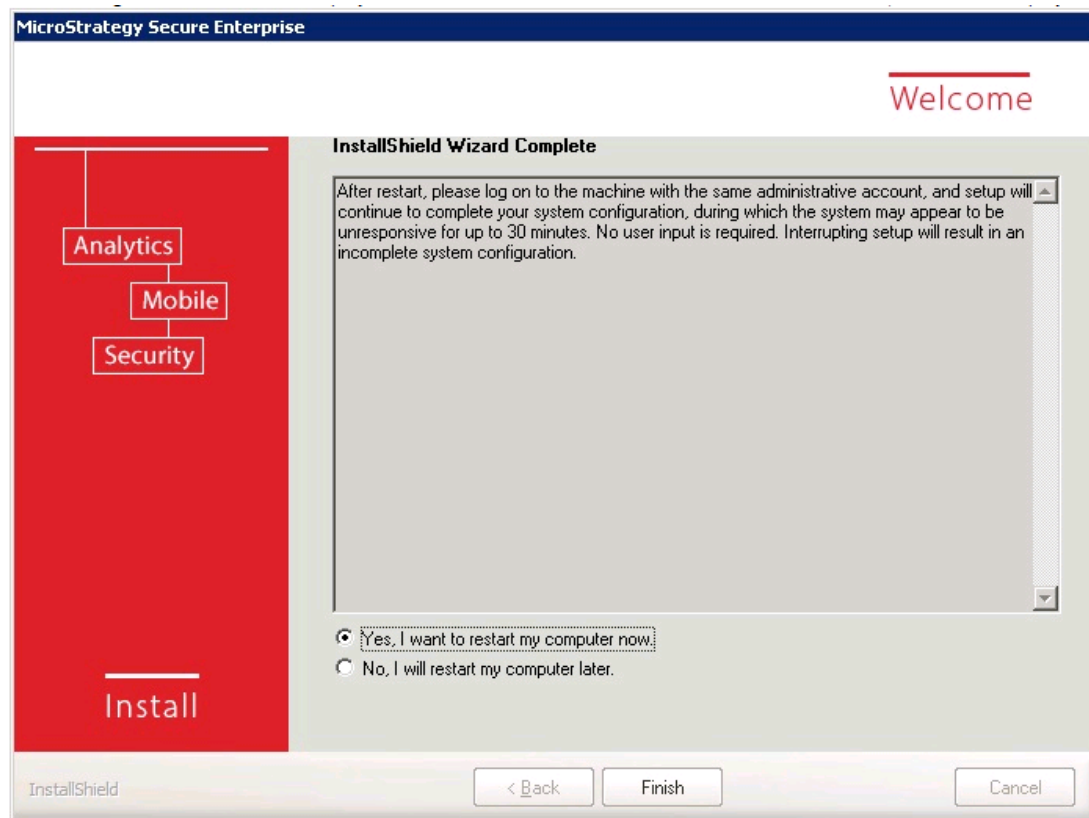
 Click **Print** to print a copy of this information for your records.

MicroStrategy Installation Wizard Complete

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

 This is the recommended procedure but you can also choose to continue without restarting.

Click **Finish**.



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 85](#).

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:
 - You have a 64-bit Windows Server: 2008R2, 2012, or 2012R2.
 - Windows 7 (64-bit) and Windows 10 (64-bit) are supported for demo purposes.
 - .Net Framework 4.
 - 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:
 - Download and extract the MicroStrategy installation package from the MicroStrategy Download Site at <https://download.microstrategy.com/>. In the extracted files, locate `MicroStrategy.exe` or `Setup.exe`.
 - Your MicroStrategy software license key is for 64-bit servers. To request a license key, go to the license key generator in the MicroStrategy Download Site at <https://software.microstrategy.com/>, contact your MicroStrategy Representative or contact MicroStrategy Technical Support at support@microstrategy.com.
 - After installation, a MicroStrategy Landing page containing links to the main MicroStrategy Platform services is displayed. The page requires JavaScript to be enabled to execute inside a web browser. Contact your IT administrator for assistance.
 - If Microsoft Internet Information Server (IIS) is present on the machine, the Express installation includes Subscription Portal and the MDX Cube Provider. If IIS is not present, these two components are not installed.
- Connectivity requirements:

- Your Windows computer must be accessible through a Fully Qualified Domain Name.
- Your mobile device must be able to connect to your Windows server. If not, you 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:

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

- **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 chain (.pem) file. You need to obtain this from a third-party signing authority, such as Verisign or Thawte. Contact your IT administrator for assistance.
 - An HTTPS/SSL server certificate (.crt) file, and the matching key (.key) file. These must match the Fully Qualified Domain Name of your 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 to 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.

- **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>.
9. You can choose to add R and R Integration Pack to your installation to have these analytics tools installed and configured. Your options are:
 - **Option 1:** Click **Next** to have the files downloaded and installed for you.
 - **Option 2:** Click the **R** link to manually download the files to your **Downloads** folder to manually install them later. See the [R Integration Pack User Guide](#) for installation procedures.
 - **Option 3:** Click **Skip** to proceed without installing R analytics or R Integration Pack.
10. You have now entered all the information required to proceed. To change settings, click **Back**.
11. Check the box marked **Automatically restart my computer when the installation completes** if you wish to enable an automatic system reboot.
12. Click **Install**. Installation takes approximately 20 minutes with the recommended hardware.
13. If you did not choose to restart automatically, restart your machine now by clicking **Finish**. The MicroStrategy software is not configured until after a restart.
14. After your machine has restarted, log in with your administrative account. Platform configuration automatically begins. This can take up to 30 minutes with the recommended hardware.
 - On Windows 7, 2008, 2012, and 2012R2, it is normal to experience a black screen, for up to 15 minutes, before a progress bar is displayed.
 - On Windows 10, the operating system loads as normal, while the configuration continues in the background. Wait for the progress bar to display before resuming normal operations.
 - A progress bar displays while the post-configuration steps are processed.
 - You will receive two emails with instructions to retrieve your Usher Security badges during this time period.
15. When the installation is complete, your default browser automatically opens the


MicroStrategy Secure Enterprise launch page.

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

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

To log in to 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).

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

2. From your mobile device:
 - Download the Usher Security application for your iOS or Android device.
 - 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:

- Click **Credentials** below the QR code.
 - Type `Administrator` for the username and keep the password empty.
 - Click to log in.
2. After logging in, you can explore the capabilities of MicroStrategy by following the guided tutorials or exploring on your own, and creating your own 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
- MicroStrategy Library

Usher Security

- Usher Server
- Usher Gateway Server
- Usher Network Manager
- Usher Analytics
- Usher Professional

Third-party software libraries

The MicroStrategy installation installs third-party software libraries to provide a streamlined user experience. For a list of the libraries and versions, see the “System Requirements” in the [Readme](#).

Ports and connectivity information

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

- Tomcat: 8080
- Network Manager: 443
- Usher Server 1-way: 1443
- Usher Server 2-way: 2443
- Gateway Server: 9501
- MySQL: 3306
- Intelligence Server: 34952
- Collaboration Server: 3000
- Export Engine Micro-Service: 20100
- Health Center Agent: 44440
- Apache ZooKeeper: 2181
- Apache Kafka: 9092
- Memcached: 11211
- Redis: 6379
- MongoDB: 27017

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=UsherCACChain.pem`
- `C:\Program Files (x86)\MicroStrategy\Usher\Usher Server\usherApps\shardGateway\conf\server.xml`
 - `SSLCertificateFile=HTTPSServerCertificate.crt`
 - `SSLCertificateKeyFile=HTTPSServerCertificate.key`
 - `SSLCACertificateFile=UsherCACChain.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=UsherCACChain.pem`

In addition,

- Certificates from `UsherCACChain.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 79](#) 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 5, Activating Your Installation](#).
- After restarting your machine to complete an initial MicroStrategy installation, the **MicroStrategy Configuration Wizard** opens. The Configuration Wizard allows you to configure your MicroStrategy production environment. For more information, refer to [Chapter 6, Configuring and Connecting Intelligence Server](#).

INSTALLING MICROSTRATEGY ON LINUX

This chapter describes the procedure for installing MicroStrategy on Linux platforms and covers the following sections:

Installation procedures on Linux	109
Configuring your MicroStrategy installation	129

Before installing MicroStrategy products, see [Chapter 1, Planning Your Installation](#) for important pre-installation information.

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

For supporting installation information, see [Installing MicroStrategy on Windows](#).

Installation procedures on Linux

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

- MicroStrategy Intelligence Server
- MicroStrategy Web
- MicroStrategy Portlets
- MicroStrategy GIS Connectors
- MicroStrategy Web Services for Office
- MicroStrategy Mobile
- MicroStrategy Command Manager
- MicroStrategy Integrity Manager
- MicroStrategy Messaging Services
- MicroStrategy System Manager

- Usher Security Services
 - Usher Security Server
 - Usher Network Manager
- Usher Analytics
- Usher Professional
- MicroStrategy Library

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.

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

For more information about these products, see *[MicroStrategy products and components, page 19](#)*.

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

- Intelligence Server
- Enterprise Manager Service
- Data Wrangling process
- Hadoop Gateway Manager
- Kafka
- Zookeeper
- Intelligence Server Log consumer
- Export Engine

It is recommended that you install MicroStrategy products as the root user.



- If you are installing MicroStrategy products with a CPU-based license, you must be logged in as the root user; otherwise an error message is displayed and the installation fails.
- If you want a non-root user to be the administrator of the server, you must manually change the ownership after running the installation. Intelligence Server operation is dependent on root user privileges and permissions. Therefore, changing the ownership of Intelligence Server to a non-root user is not a certified or recommended practice.
- Only a user with root permissions can install 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, see [Usher Pre-Installation Instructions](#).

- 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 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 Linux graphical user interface. This also lets you perform an installation on other machines.

For information on how to perform a silent installation on a Linux environment, see [Chapter 10, Automated Installation on Linux](#).

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 Linux account. For ease of management and maintenance, it is recommended that you create a dedicated user account.



- You need root access permissions for installation if you have purchased the CPU-based MicroStrategy license.
- If you want to enable additional error and troubleshooting issue logging for the MicroStrategy installation routine, contact MicroStrategy Technical Support.

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

To access the MicroStrategy Installation Wizard

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

2 Browse to the MicroStrategy Installation folder and then `QueryReportingAnalysis_Linux`



- You can access the installation files by asking your system administrator to share the files on a network location. There are different installation files for installing MicroStrategy products on different platforms; Windows and Linux.
- You can reduce the amount of data that has to be downloaded for the installation by excluding all of the `.tzip` 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 77](#). Details on using a `options.txt` file to provide the location of the installation files are provided in [Chapter 10, Automated Installation on Linux](#) and the parameters used to specify the location of the required installation files are described in [Providing installation files for smaller installations, page 347](#).

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 330](#).

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



- To complete the installation, you must have write permissions in the installation directory; otherwise the installation fails.
- After you enter all required information on an installation page, click **Next**, or press 1 and then press `ENTER`, to proceed to the next page of the installation routine.
- To quite the installation at any time during the setup, click **Cancel**, or press 3 and then press `ENTER`.

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 on the current machine. The steps provided here assume that you are either installing for the first time or creating a new installation.

You can support multiple installations of MicroStrategy on Linux machines. Additionally, you can also modify, repair, and upgrade existing MicroStrategy installations. This 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 `$HOME/MicroStrategy` if you do not have write access to `/var/opt/MicroStrategy`.
 - Do not install the MicroStrategy configuration files directly to your Linux Home Directory (`$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 Professional, components of MicroStrategy Analytics Enterprise are also installed. To install Usher Professional, 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 19](#) 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:

Installation File	MicroStrategy Components That Require The Installation File
mstr1.tzp	All MicroStrategy components and products
mstr3.tzp	MicroStrategy Intelligence Server and all of its components
mstr4.tzp	MicroStrategy Web, including Web Analyst, Web Reporter, and Web Professional
mstr5.tzp	MicroStrategy Web Services for Office
mstr6.tzp	MicroStrategy Command Manager
mstr7.tzp	MicroStrategy Integrity Manager
mstr8.tzp	MicroStrategy System Manager
mstr9.tzp	MicroStrategy Mobile Client
mstr10.tzp	MicroStrategy Mobile Server

Installation File	MicroStrategy Components That Require The Installation File
mstr11.tzp	MicroStrategy Portlets, which is a component of MicroStrategy Web
mstr12.tzp	MicroStrategy GIS Connectors, which is a component of MicroStrategy Web
mstr13.tzp	All MicroStrategy components and products
mstr15.tzp	MicroStrategy Usher Server
mstr16.tzp	MicroStrategy Usher Network Manager
mstr17.tzp	MicroStrategy Usher Analytics
mstr18.tzp	MicroStrategy Usher Professional

- Provide the location of the installation files using an `options.txt` file. This lets you access the installation files stored on a folder or stored at a URL and accessed using HTTP or HTTPS. Details on using an `options.txt` file as part of an installation are provided in [Chapter 10, Automated Installation on Linux](#) and the parameters used to specify the location of the required installation files are described in [Providing installation files for smaller installations, page 347](#).

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 about this setting and the recommendation for its value, see [Supporting Intelligence Server memory allocation on Linux, page 63](#).

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 see the limit recommendations in [Configuring shared memory resources, page 61](#).

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 61](#).

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. Health Center can email a system administrator if it encounters a problem. 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 about configuring and using Health Center, see 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

Specify the following configuration information to define the machine as 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.



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

Contact Information

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

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



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

Request Activation Code

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

- Select **Yes, I want to request an Activation Code** and click **Next** to request an Activation Code. The Activation Code is sent to the email addresses supplied in the Installer Information and Contact Information pages.
- Select **No, I will request the Activation Code at a later time** and click **Next** to request an Activation Code at a later time.

If you choose to request an Activation Code at a later time, a message is displayed that instructs you how to request an Activation Code after the installation procedure is completed. For more instructions on requesting an Activation Code at a later time, see [Request an Activation Code, page 148](#) in [Chapter 5, Activating Your Installation](#).

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

Once you request an Activation Code, an email is sent to the email addresses you specify in the Installer Information and Contact Information pages of the software activation procedure. The email provides instructions on how to use the requested Activation Code to activate your software. To activate your installation, you can also use the steps given in [Activate your installation, page 150](#) in [Chapter 5, 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 1443.
- **Client and Server (two-way SSL) mutual authentication:** The default port is 2443.

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

- **SSL Certificate File:** The Usher Server SSL/HTTPS certificate is used to encrypt the data and enforce authentication. The file contains the certification part of the signed certificate, as well as its public key, in a crt format. By default, the path is `USHER_SERVER_INSTALL_PATH/usherApps/shardIDM/conf/Server_ca.crt`.
- **Private Key File:** The Usher Server SSL/HTTPS Private Key file is used to decrypt the data and enforce authentication. The file contains the key part of the signed certificate in a key format. By default, the path is `USHER_SERVER_INSTALL_PATH/usherApps/shardIDM/conf/Server_ca.key`.
- **SSL Certificate Chain:** The Usher Server Certificate Authority Chain includes all the certificate authority (CA) certificates that the Usher server will trust. The file contains the information in PEM format. By default, the path is `USHER_SERVER_INSTALL_PATH/usherApps/shardIDM/conf/Server_ca.pem`.

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

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).

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 6, Configuring and Connecting Intelligence Server](#).

Click **Finish** to complete the installation.

Unique post-installation configurations

MicroStrategy supports many different Linux environments with various system configurations. There are a few cases in which you must perform some manual configurations to support the use of MicroStrategy on your system.

- [Migrating Intelligence Server from Windows to Linux, page 125](#)
- [Create links for Intelligence Server startup in SUSE Linux, page 126](#)
- [Supporting fonts for documents, exported reports, and graphs, page 126](#)
- [Starting and Stopping the PDF Exporter Service](#)

Migrating Intelligence Server from Windows to Linux

If you are installing MicroStrategy Intelligence Server on 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 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:

```
ln libssl.so.0.9.7 libssl.so.4
```

- 3 In a console window, create the link of `libcrypto.so.4` to `libcrypto.so.0.9.7` with the following command:

```
ln libcrypto.so.0.9.7 libcrypto.so.4
```

Supporting fonts for documents, exported reports, and graphs

When Intelligence Server is running on a Linux platform, all fonts are converted to the Courier New font for:

- Reports exported to PDF format
- Report Services documents
- Graphs contained in HTML documents
- Graphs displayed in MicroStrategy Web

This occurs because the fonts required by the PDF component are missing from Linux machines running Intelligence Server.



MicroStrategy cannot package these fonts with Intelligence Server due to licensing restrictions.

For steps to support fonts such as Microsoft True Type fonts for the MicroStrategy features listed above, see [Setup for executing Report Services documents, page 279](#).

Starting and Stopping Intelligence Server Kafka service

For Linux users, the Kafka Consumer will be installed and automatically running after installation or upgrade. You will find `KafkaConsumer.sh`, `LogConsumer.properties`, and `KafkaConsumer.jar` under `<install_`

`path>/IntelligenceServer/KafkaConsumer`. You can use the following commands to control the Kafka Consumer logging activity:

- **Start service:** `<path_to>/KafkaConsumer.sh start`
- **Stop service:** `<path_to>/KafkaConsumer.sh stop`
- **Restart service:** `<path_to>/KafkaConsumer.sh restart`
- **Check service status:** `<path_to>/KafkaConsumer.sh status`

Starting and Stopping the PDF Exporter Service

For Linux users the PDF Exporter Service will not be started when OS is restarted. To start this service manually, run the `pdfexporter.sh` script found under the path `install/IntelligenceServer/PDFExportService`. Commands for this script include:

- `<path_to>/pdfexporter.sh start` to start the PDF Exporter Service.
- `<path_to>/pdfexporter.sh stop` to stop the PDF Exporter Service.
- `<path_to>/pdfexporter.sh restart` to restart the PDF Exporter Service.
- `<path_to>/pdfexporter.sh status` to check the status the PDF Exporter Service.

For other configuration settings, see [Export Engine configuration](#).

Verifying installation

During installation, the MicroStrategy Installation Wizard gathers and records information about your system and your installation selections. You can verify the setup information through the installation log file (`install.log`).

By default, the log file is located in `INSTALL_PATH` where, `INSTALL_PATH` is the directory you specified as the install directory in the MicroStrategy Installation Wizard.

The log file includes information about the following:

- Installation date
- Target directory
- Operating system identification
- Selected installation options
- Selected licensing details



This log file can be helpful if you encounter errors during the installation process. The log file records the reasons due to which the errors occurred.

Directory structure

The following table lists the directories in which MicroStrategy files are installed.

PATH/Directory	Contents
<i>HOME_PATH</i>	Configuration files that can be modified after installation.
<i>HOME_PATH/env</i>	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.
<i>HOME_PATH/bin</i>	Scripts to launch the MicroStrategy applications.
<i>INSTALL_PATH</i>	Files that are not supposed to change after the installation is complete.
<i>INSTALL_PATH/CommandManager</i>	MicroStrategy Command Manager files. This is the default directory for Command Manager but another location can be selected during installation.
<i>INSTALL_PATH/Help</i>	Documentation and Help for MicroStrategy products.
<i>INSTALL_PATH/GISConnectors</i>	MicroStrategy Portlet files. This is the default directory for the Portlets but another location can be selected during installation.
<i>INSTALL_PATH/IntelligenceServer/bin</i>	Intelligence Server-specific binary files.
<i>INSTALL_PATH/Mobile</i>	MicroStrategy Mobile and Mobile Server JSP files. This is the default directory for Mobile but another location can be selected during installation.
<i>INSTALL_PATH/PDFGeneratorFiles</i>	Support files (fonts) for the PDF generation feature of Intelligence Server.
<i>INSTALL_PATH/ReleaseNotes</i>	<i>MicroStrategy Readme</i> for this release of MicroStrategy products.
<i>INSTALL_PATH/Portlets</i>	MicroStrategy Portlet files. This is the default directory for the Portlets but another location can be selected during installation.
<i>INSTALL_PATH/SystemManager</i>	MicroStrategy System Manager files. This is the default directory for System Manager but another location can be selected during installation.
<i>INSTALL_PATH/Usher</i>	MicroStrategy Usher files.
<i>INSTALL_PATH/WebServicesJ2EE</i>	MicroStrategy Web Services deployment path.
<i>INSTALL_PATH/WebUniversal</i>	MicroStrategy Web deployment path.

PATH/Directory	Contents
<i>INSTALL_PATH/_jvm</i>	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.
<i>INSTALL_PATH/_uninst</i>	Launch files for uninstalling MicroStrategy.
<i>INSTALL_PATH/bin</i>	64-bit binary files.
<i>INSTALL_PATH/bin32</i>	32-bit binary files.
<i>INSTALL_PATH/help</i>	Online help files.
<i>INSTALL_PATH/jar</i>	Java libraries.
<i>INSTALL_PATH/lib</i>	64-bit binary libraries.
<i>INSTALL_PATH/lib32</i>	32-bit binary libraries.
<i>INSTALL_PATH/locale</i>	ODBC support messages.
<i>LOG_PATH</i>	MicroStrategy application log files, which includes Intelligence Server log files.

Configuring your MicroStrategy installation

After completing the steps to install MicroStrategy products, you can set up and configure your installation. To help guide the rest of your installation and configuration steps, refer to the section [Installing and configuring MicroStrategy on Linux, page 80](#) in [Chapter 1, Planning Your Installation](#), for an installation and configuration checklist.

To configure the Usher components, refer to the [Usher Help](#).

INSTALLING AND CONFIGURING USHER

Usher Security includes the following MicroStrategy products:

- Usher Security Platform (which includes Usher Network Manager and Usher Server)
- Usher Analytics
- Usher Professional

You can install Usher Security as part of the entire MicroStrategy Secure Enterprise or by itself. If you chose to install Usher Security by itself, be aware that certain MicroStrategy components are required for Usher Analytics and Usher Professional and these components are installed in the background.

If you are upgrading an existing MicroStrategy Usher install, see the [Upgrade Guide](#).

Overview of Usher Install

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

- **Step 1: [Pre-Installation](#)**
Prepare for the installation by verifying that you have the required information and configuring the server with the necessary software for Usher.
- **Step 2: [Installation](#)**
Perform the installation.
- **Step 3: [Post-Installation](#)**
Configure Usher for your enterprise environment.

For assistance with common issues, see [Troubleshooting Information](#). If you require additional assistance, contact [Technical Support](#).

Usher Pre-Installation Instructions

To prepare for the installation, you must verify that you have the required information and that the server is configured with the necessary software for Usher. Review the following sections that apply to your environment and perform the necessary actions to satisfy all prerequisites before the installation

Common Prerequisites (Windows and Linux)

- You have an installation key.
- Setup the server with a fully qualified domain name (FQDN).
- Obtain the necessary SSL files.
- Open the required ports in the firewall. See [Default Usher Communication Ports](#).
- If SMTP requires SSL connection, obtain SMTP file server information: hostname, port, and user credentials for designated User Server administrator.
- You have verified that the install environment includes the necessary software. See the “System Requirements” section of the [Readme](#).

To verify the version of the operating system:



- Windows: Run following command in a Command Prompt window: `winver`
- Linux: Run the following command: `cat /etc/*-release`.

Windows Prerequisites

- **MySQL Open Source Components:** The Usher installation relies on MySQL open source components that are not provided in the MicroStrategy installer. The required components can be downloaded during the installation. However, if the server does not have Internet access, you must manually copy the following files to the user's Downloads folder:
 - **MySQL:**
<http://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.28-winx64.zip>
 - **MySQL Connector/ODBC 5.3.4:**
<http://dev.mysql.com/get/Downloads/Connector-ODBC/5.3/mysql-connector-odbc-5.3.4-winx64.msi>
 - **MySQL Connector/Java 5.1.22:**
<http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.22.zip>
 - **MySQL Connector/Python 2.1.3:**
<http://dev.mysql.com/get/Downloads/Connector-Python/mysql-connector-python-2.1.3-py2.7-winx64.msi>
 - **MySQL time zone description tables:**
http://downloads.mysql.com/general/timezone_2015g_posix.zip

Linux Prerequisites

- You have Linux root user permission to complete the pre-installation steps.
- If you are configuring Usher Analytics and Usher Professional after installing Usher Security, you need to install Python 2.7.
- Do not save files in the `\root\` folder path. Doing so will prevent the successful install of Usher Security.

Pre-installation steps

Disable SELinux

- 1 Disable SELinux by navigating to:

```
/SELinuxinstallpath/selinux/config
```

In the configuration file, change:

```
SELINUX=enforcing to SELINUX=disabled
```

- 2 Verify that SELinux has been disabled:

```
sestatus
```

- 3 Restart the server to make the changes permanent.

Set up Apache HTTP server

To verify the version of the Apache HTTP server, enter the following command: `> httpd -v`

If you do not have the Apache HTTP server installed, enter `yum install httpd`, then enter the following command to start the service:

```
> service httpd start
```

Download and set up Java Developer Kit

- 1 Verify whether Java DK is 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 Start the installation process and follow the prompts.
 - 6 To verify that the JDK was successfully installed, enter the following commands:
 - `> which java`
 - `> java -version`
-

Download and set up Apache Tomcat



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

- 1 Navigate to `http://tomcat.apache.org/download-80.cgi`, and under Core in Binary Distributions, download the `tar.gz` file.
 - 2 Install and follow the prompts, noting the location of the download location for Tomcat.
 - 3 Install third-party libraries into the Tomcat directory, and note the location of the libraries. Open each of the following websites, download the specified files, and install the files in the Tomcat library:
 - a From `http://www.xom.nu`, download the complete `tar.gz` file. Install the file in the Tomcat library.
 - b From `http://logback.qos.ch/dist/`, download both the 1.1.2 `tar.gz` core and classic files. Install the files in the Tomcat library.
 - c From `https://jcifs.samba.org/`, download the 1.3.17 `.jar` file. Install the file in the Tomcat library.
 - d From `https://sourceforge.net/projects/wsdl4j/`, download the `wsdl4j 1.6.2.jar` file. Install the file in the Tomcat library.
 - e From `http://dev.mysql.com/downloads/connector/j/`, download the `connector/j 5.1.xxtar.gz` file. Install the file in the Tomcat library.
 - 4 Verify the Tomcat install:
 - a Start Tomcat: `> cd /(tomcat8)/bin`
 - b Display the Tomcat browser: `> ./startup.sh`
 - c In a web browser, navigate to: `http://localhost:8080`
 - 5 Install Apache Tomcat Native Library. See [Tomcat official documentation](#) for detailed information.
-

Set up the timezone on Linux

Set the system timezone to `GMT localtime`.

Enter the following commands:

- `> cd /etc`
- `> rm localtime`
- `> ln -s /usr/share/zoneinfo/GMT localtime`

Set up the Fully Qualified Domain Name

The following procedure requires you to work with your IT department to change the hostname. Additionally, you must obtain a trusted vendor CA-signed certificate, as well as generate a self-signed certificate. Refer to your company's IT policies for this section.

- 1** Contact your IT department for information about your company policy on Fully Qualified Domain Names (FQDN).
- 2** Change the Linux hostname to a FQDN, per your company's policy. Then have the new FQDN and IP address mapped.
- 3** Following the direction of your IT department to obtain a CA-signed Usher Server SSL/HTTPS certificate from a trusted vendor.

If you are asked for a certificate request that contains your company information, perform the following steps:

- a Create and submit a certificate request (`*.csr`) and private key (`*.key`) to the vendor in the form requested.
 - b Provide your company information to the certificate vendor. This information is incorporated into your certificate request.
- 4** Upon receiving your Usher Sever SSL/HTTPS Certificate and 1 or more Certificate Authority Certificates from the third-party vendor, create a Certificate Authority Chain (`*.pem`) file by combining these certificates (`*.crt`) files. Enter the following command:
 - `> cat <PathToCertFolder>/rootCA.crt
<PathToCertFolder>/intermediateCA.crt
<PathToCertFolder>/HTTPSServerCertificate.crt >
<PathToCertFolder>/UsherCAChain.pem`
 - 5** Create a self-signed Usher signing CA certificate and append it to the CA chain. See [Managing the Usher Signing Certificate Authority](#).

Install memcached

Optionally, you can install memcached, which will increase the speed of transactions. After installing memcached, you must configure the modules for Usher Security.

- 1** Install memcached by using either of the following methods:
 - For **yum**, use the following command: `> yum install memcached php-pecl-memcache.`

- For **rpm**, use the following command: `> rpm -i memcached-x.x.x-x.el6.x86_64.`

2 Configure memcached by changing the following in the memcached file:

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

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

4 Verify that the server is running by verifying the status.

Download and set up PHP



You may encounter problems while setting up PHP. To troubleshooting error messages, see your IT department and third-party reference material.

1 Use your subscription manager to enable SCL:

- `subscription-manager repos --enable rhel-server-rhsc1-6-eus-rpms`

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

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

3 Install the updates database module for MySQL:

- `yum install php54-php-mysqldb`

4 Disable the loading of php 2.3 Apache:

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

5 Restart Apache to make the changes:

- `service httpd restart`

6 After the installation is complete, enable `mod_rewrite`. To do this, enter `/InstallPath/etc/httpd/conf/httpd.conf` to modify the file. The parameters are as follows:

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

</Directory>

7 Verify that the required PHP modules have been installed correctly:

a Enter the following commands:

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

b Execute the following command to add `/etc/httpd/conf/httpd.conf`:

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

c Enter the following commands to create `test.php` in `/var/www/html`:

- `<?php`
- `phpinfo();`
- `?>`

d Restart the service to complete the changes.

e In a web browser, navigate to `http://localhost/test.php`. You should see the PHP landing page, which signals that the verification is successful.

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. Even if you already have MySQL installed, 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, navigate to <https://dev.mysql.com/downloads/mysql>. From the drop-down list, select the correct platform of MySQL Community Server for your system. Find your operating system version, and download the RPM file.
- 2** Navigate to <http://dev.mysql.com/downloads/repo/yum/>, and download the correct RPM package for your system.
- 3** Open a command window and begin the installation by entering the following commands:
 - For MySQL yum repository, enter: `> yum localinstall mysql-community-release-el6-5.noarch.rpm`
 - For MySQL Community Server, enter: `> yum install mysql-community-server`
 - For MySQL JDBC connector, enter: `> yum install mysql-connector-java`

After installation, you must set the classpath for the JDBC connector, which depends on the type of shell you are running:

- For bourne-compatible shells, enter: `> export CLASSPATH=/home/user/mysql-connector-java-5.1.34-bin.jar:$CLASSPATH`
 - For C shell, enter: `> setenv CLASSPATH /home/user/mysql-connector-java-5.1.34-bin.jar:$CLASSPATH`
 - For MySQL ODBC connector, enter: `> yum install mysql-connector-odbc`
 - For MySQL python library, enter: `> yum install mysql-python`
 - For MySQL PDO extension, enter: `> yum install with-pdo-mysql`
- 4** Enter: `> mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -h127.0.0.1 -P3306 -uroot -p mysql` to install the `convert_tz` functions.



If an error message displays indicating that the function is unable to load, verify that the function has been properly installed. To do this, enter the following command: `> select convert_tz (current_timestamp, 'utc', 'est`. If the file is timestamped, then it is installed.

- 5** To complete the installation process, restart MySQL. Enter:


```
service mysqld start
```
- 6** Set the root password for MySQL.
 - 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 To create the mstr user, enter: `> create user 'username'@'localhost' identified by 'password';`
 - b To grant the mstr user access, enter: `> grant all on *.* to 'username'@'localhost' with grant option;`
- 8 If you intend to use Usher with MySQL 5.7.4 and above, the following line needs to be added or modified in the MySQL `my.cnf` file:


```
sql-mode="STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION".
```
- 9 If you intend to implement Usher Analytics, install crontab for MySQL.
 - a Verify that this directory exists: `> /usr/bin/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`

Usher Installation Instructions

After completing all the [pre-installation](#) steps, you are ready to begin the installation.

For installation instructions, see the following:

- **Windows:** [Installing MicroStrategy on Windows](#)
- **Linux:** [Installing MicroStrategy on Linux](#)



Be sure to perform all prerequisites and to note Usher-specific information in the installation instructions.

After completing the installation, you must perform the [Usher Post-Installation Instructions](#).

Usher Post-Installation Instructions

After you have installed the Usher software, you must deploy and complete the configuration of the Usher Server and Network Manager:

- [Deploy the Usher Server](#) (Linux)
- [Enable HTTPS for Usher Network Manager](#) (Linux)

- [Complete the Usher Configuration](#) (Windows and Linux)
 - [Set up Usher Gateway Server](#) (Windows and Linux)
-

To deploy your Usher Server (Linux)

- 1 In a command window, navigate to the installation location of the 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, see 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:

```
> cp /<localdirectory>/apache-tomcat-8.0.26/lib/logback-  
classic-1.1.2.jar  
/<  
serverpath  
>/Usher/UsherServer/usherApps/shardIDM/webapps/ROOT/WEB-  
INF/lib
```

- 4 Restart the Usher Server to make the changes permanent. Enter the following commands:

- ```
> cd
/<serverpath>/Usher/UsherServer/usherApps/shardIDM/bin
```
- ```
> ./tomcat.sh restart
```

- 5 Verify that the Usher Server has restarted. In a web browser, navigate to:

```
https://<FQDN>:<1-way port>
```

where

- `<FQDN>` is your domain name
- `<1-way port>` is your SSL one-way port number (1443 is the default one-way port)

- 6 After the Usher server is deployed, navigate back to the 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 (Linux)

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:


```
vi /etc/selinux/config
```
- 2 Replace disabled


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

To enable HTTPS

- 1 Install `ssl_module` by entering the following command:


```
> yum install mod_ssl
```
- 2 Verify that the module was enabled during the installation:


```
> httpd -M | grep ssl_module
```
- 3 Using Vim or any other file editor, open the Network Manager configuration located at `/etc/httpd/conf.d/manager.usher.com.conf` and update the necessary file parameters.

The file should look similar to the following:

```
Listen 443 # Verify both ports are the Network Manager port
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile <PathToCertFolder>/HTTPSServerCertificate.crt # Verify
this path is valid
    SSLCertificateKeyFile <PathToCertFolder>/HTTPSServerCertificate.key #
Verify this path is valid
    SSLCACertificateFile <PathToCertFolder>/UsherCAChain.pem # Add this path
if not already present
    SetEnvIf Usher-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
    Alias /networkmanager
/usher/Microstrategy/install/Usher/UsherNetworkMgr/networkmanager
    # Verify this path is valid
    <Directory
/usher/Microstrategy/install/Usher/UsherNetworkMgr/networkmanager> # Verify
this path is valid
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Save and close the file.

To complete the Usher Configuration (Windows and Linux)

Prerequisites

You need the following to complete the Usher configuration:

- SSL port numbers from your MicroStrategy installation steps
- Usher Server SSL/HTTPS Certificate (.crt) and Private Key (.key) files
- Usher Signing Certificate (.crt) and Private Key (.key) files
- SMTP server
- Google Maps API key (if you plan to use Usher Network Manager and Google Maps)

- 1 Open the Usher Network Manager page by entering the following in a web browser:

`http<s>://<FQDN>:<port>/networkmanager/managesystem`

- 2 Enter your Usher Server URL and one-way port, making sure you use your FQDN (for example, `https://<FQDN>:1443`), and click **Enter**.



If you enter an IP Address instead of your FQDN, you will not be able to proceed.

- 3 In the **License Key** field, provide your installation key (universal or 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:** A description of the Usher Server instance as it will appear in the Usher Security app.
- **Usher File Directory:** Keep the default value.
- **SSL Certificate Authority Certificate:** Your self-signed .crt certificate that was generated using OpenSSL.
- **SSL Certificate Authority Key:** Your self-signed *.key that was generated using OpenSSL.
- **SSL Certificate Authority Key Password:** If you have a password for your key, select the **Required** check box, and then enter your password. If you did not assign a password, leave this field blank.
- **AES Key:** Your encryption key generated during the Usher Server installation. Keep the default value.
- **SAML Certificate:** Your trusted vendor CA-signed .crt.
- **SAML Key:** The key for the trusted vendor CA-signed certificate.

- **SMTP Server:** Your company's SMTP server.



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

- **Port:** Your company's SMTP server port.
- **SMTP Authentication:** If your server is password protected, select the **Required** check box. Enter your username and password in the corresponding fields.
- **SMTP Configuration:** Leave this field blank.
- **Email Sender Address:** The email address that you are using to send the badge invitations for the Usher network.
- **Security Server Host:** Keep the default URL and enter the Usher Server two-way SSL Port value (default value is 2443). For example, `<FQDN> : 2443`. 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, `<FQDN> : 9501`. For the Gateway Host, `https://` is not needed.
- **Gateway Load Balancer:** Keep the default URL and enter your Agent Gateway port value to match the values for Gateway Host.
- **Google Maps API Key:** If you are using Google Maps for your Usher Network, enter the third-party key.
- **Memcached:** No node is required for the Usher configuration.
- **iPad Configuration Link:** If you have already set up Usher Professional, enter your iPad URL link. This link comes from the MicroStrategy Mobile configuration and begins with "mstrusheripad".
- **iPhone Configuration Link:** If you have already set up Usher Professional, enter your iPhone URL link. This link comes from the MicroStrategy Mobile configuration and begins with "mstrusher".
- **Support Email:** The email address that will be displayed in the Usher Security app and used as the default address to send support emails.
- **Support Phone Label:** The label for the support number that will be displayed in the Usher Security mobile app. The default is "Phone Number".
- **Support Phone:** The support number that will be displayed in the Usher Security mobile app.
- **Push Notification-based Capabilities:** This service is not required for the Usher configuration.

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.
 - **Certificate Path:** Leave this field blank.
 - **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.
 - **Help Page Base URL:** Leave the default value.
 - **Plugin Host Server:** Keep the default value.
 - **Google Client ID:** If you have a business Google Drive account, enter your Google ID number.
 - **Google Client Secret:** If you want to import users from your Google Drive account, enter their information.
 - **Salesforce Client ID:** If you have an executive Salesforce account for your company, enter your Salesforce ID Number.
 - **Salesforce Client Secret:** If you want to import users from your Salesforce account, enter their information.
- 7** To save your changes and complete the configuration, click **Done**.
- 8** Restart the service:
- Windows:**
- From the **Start** button, choose **Administrative Tools > Services**. Right-click on the **Apache Tomcat 8.0 shardIDM** service and select **Restart**.
- Linux:**
- ```
> cd
/<serverpath>/Usher/UsherServer/usherApps/shardIDM/bin

> ./tomcat.sh restart
```
- 9** Click **Create an Usher Admin**. If you need to return to this step later, type the following into the browser:  
`http<s>://<FQDN>:<port>/networkmanager/firstUA/create`.
- 10** To upload a photo for the Admin account, click to select a file. Supported image formats are .png, .jpeg, and .jpg.
- 11** Enter your first and last names in the correct fields.
- 12** In the **Email Address** field, type the email address to associate with your Usher Admin account.
- 13** Click **Create**. Usher sends an email invitation to the email address that you provided.
- 14** Open the email in your smartphone, then click **Get My Badge**, which opens the Usher Security application.

- 15 In a web browser, navigate to the Network Manager home page `http<s>://<FQDN>:<port>/networkmanager`, and scan the QR code displayed.
- 16 After you log in, you can create security networks and add users. For more information, see the [Usher Online Help](#).

If you want to create additional Usher Administrators, see [Adding and deleting Usher Administrator](#).

If you want to synchronize Usher with an existing IDM, see [Synchronizing users from Microsoft Active Directory](#).

---

## To set up your Usher Gateway Server (Windows and Linux)

---

After you have configured Usher Server and Network Manager, deploy the Gateway.

### For Linux:

- 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.

### For Windows and Linux:

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

```
https://<FQDN>:<port>/gateway/test/.
```
- 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, see the [Usher Help](#).

## Troubleshooting Information

For Usher administration processes, see [Usher Administration](#).

### Installation Error Codes

For an explanation of the error codes (0 - 24) displayed by the Linux installer, see [TN300224](#).



## Default Usher Communication Ports

When using MicroStrategy Usher, you need to ensure that certain ports are available. The default ports are:

- **80/443** - the port used by Usher Network Manager
- **1443** - an SSL-enabled port used for client-server communication with the Usher Server
- **2443** - an SSL-enabled port for two-way (mutual) authentication with the Usher Server
- **9501** - an SSL-enabled port used for communication with the Usher Gateway Server

## SSL Certificate Verification

You can verify your SSL certificates using the following command:

```
openssl verify -CAfile file [path_to_your_pem] [path_to_your_cert]
```

For details, see the [OpenSSL documentation](#).

## Usher Logs

The Usher components log information about their operations. When troubleshooting issues, you can review the logs for each component.

### Database logs

Before the database is created, during installation and configuration, error messages related to Network Manager may be found in **usher\_network\_manager.log**.

When Usher is installed, the MySQL instance includes a schema **usher\_server\_log** which includes logs for activities performed through Usher Security Server. The logs for Usher Network Manager are located in a different schema and table: **usher\_network\_manager.usher\_network\_log**.

### Server-side logs

The following is a list of server-side logs and their default locations. If Usher is not installed in the default location, use the folder descriptions to determine where to look.

Usher automatically archives many of its server logs on a daily basis. Older logs are located in the same folder, but the name will include the date.

- **Usher Security Server Tomcat log**
  - **Location:** Usher Security Server Tomcat logs folder
    - **Windows:** C:\Program Files (x86)\MicroStrategy\Usher\Usher Server\usherApps\shardIDM\logs\catalina.out

- **Linux:**  
/opt/MicroStrategy/Usher/UsherServer/usherApps/shardIDM/logs/catalina.out
- **Usher Security Server API log**
  - **Location:** Usher Security Server Tomcat logs folder
    - **Windows:** C:\Program Files (x86)\MicroStrategy\Usher\Usher Server\usherApps\shardIDM\logs\info.log
    - **Linux:**  
/opt/MicroStrategy/Usher/UsherServer/usherApps/shardIDM/logs/info.log
- **Usher Gateway Server Tomcat log**
  - **Location:** Usher Gateway Server Tomcat logs folder
    - **Windows:** C:\Program Files (x86)\MicroStrategy\Usher\Usher Server\usherApps\shardGateway\logs\catalina.out
    - **Linux:**  
/opt/MicroStrategy/Usher/UsherServer/usherApps/shardGateway/logs/catalina.out
- **Usher Gateway Server API log**
  - **Location:** Root Tomcat logs folder (Windows) or Usher Gateway Server Tomcat logs folder (Linux)
    - **Windows:** C:\Program Files (x86)\Common Files\MicroStrategy\Tomcat\apache-tomcat-8.0.30\logs\gateway.log
    - **Linux:**  
/opt/MicroStrategy/Usher/UsherServer/usherApps/shardGateway/logs/gateway.log

## Usher Security app logs

When an error appears in the Usher Security mobile app, you can trigger the app to create an email with the log file as an attachment.

## To report an issue with an Usher Security app log

- 1 In the Usher Security app, tap the **Settings** icon in the lower-right.
- 2 Scroll down to the “Contact Us” section and tap **Report a Problem**.
- 3 From the pop-up menu, tap the most relevant category.
- 4 In the **Let us know what happened** field, describe the issue.
- 5 Tap **Send**.

- 6 Your smartphone's email client launches automatically with an email draft addressed to the email address identified as the support contact for your Usher instance. The subject and body are pre-populated, and the app log is included as an attachment.
- 7 If necessary, change the recipient in the **To:** field or add additional recipients.
- 8 Send the email. Recipients can open the log file in a text editor to view the contents.

## Frequently Asked Questions

### General

- **What do I do if a smartphone with an Usher badge is lost or stolen?**

No problem. An Usher administrator can revoke all Usher privileges for a specific user/phone instantly.

### Usher Gateway

- **Why does nothing appear when I access the Usher Gateway?**

Typically this indicates that the Gateway Server properties are not completely configured. See [To complete Usher Configuration \(Windows and Linux\)](#).

### Usher Professional

- **Usher Professional is not working. What could be the issue?**

Verify that the **iPad Configuration Link** and **iPhone Configuration Link** fields are set correctly. The **iPad Configuration Link** must start with "mstrusheripad" and the **iPhone Configuration Link** must start with "mstrusher"., as described in [To complete the Usher configuration \(Windows and Linux\)](#). The

For example:

```
mstrusheripad:/?url=https%3A%2F%2Fenv-32205.customer.cloud.microstrategy.com%3A443%2FMicroStrategyMobile%2Fservlet%2FtaskProc%3FtaskId%3DgetMobileConfiguration%26taskEnv%3Dxml%26taskContentType%3Dxmlanf%26configurationID%3De7758ac7-cfc5-427e-9c0f-ec6bfc14943&authMode=1&dt=2
```

```
mstrusher:/?url=https%3A%2F%2Fenv-32205.customer.cloud.microstrategy.com%3A443%2FMicroStrategyMobile%2Fservlet%2FtaskProc%3FtaskId%3DgetMobileConfiguration%26taskEnv%3Dxml%26taskContentType%3Dxmlanf%26configurationID%3D6e1964bf-a04c-43c2-b6b2-aac004877fc2&authMode=1&dt=1
```

# 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 148](#)
- [Activate your installation, page 150](#)

For answers to commonly asked questions about server activation, see [Server Activation FAQ, page 151](#).

## Request an Activation Code

You can request an Activation Code by supplying MicroStrategy with important information related to your installation. The information you provide helps MicroStrategy understand how you plan to use MicroStrategy software. With this information MicroStrategy can provide better information and technical support for your software configuration.

On Windows, MicroStrategy products can be activated only in graphics user interface (GUI) mode, using License Manager.

MicroStrategy products can be activated on Linux, either in GUI mode or in command line mode, using License Manager. In both cases, License Manager runs and requests the same information. The main differences are in how you provide the information and navigate through the Activation Code request steps.

---

### Request an Activation Code with License Manager

---

If you requested an Activation Code during installation you can skip this procedure and activate your installation by following the instructions in the next procedure, [Activate your software installation, page 150](#).

**1** Open MicroStrategy License Manager:

- **Windows:** From the **Start** menu, point to **Programs**, then **MicroStrategy Tools**, and then select **License Manager**. License Manager opens.
- **Linux:** License Manager can be run in GUI mode or command line mode.
  - **GUI:** In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `bin` and type `./mstrlicmgr`, then press `ENTER`. The MicroStrategy License Manager opens in GUI mode.
  - **Command line:** In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `bin` and type `./mstrlicmgr -console`, then 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.
- **Linux:** License Manager can be run in GUI mode or command line mode.
  - **GUI:** In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `bin` and type `./mstrlicmgr`, then press `ENTER`. The MicroStrategy License Manager opens in GUI mode.
  - **Command line:** In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `bin` and type `./mstrlicmgr -console`, then 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 79* 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
- MicroStrategy information:
  - Install Type – new/modify/upgrade
  - Products and version installed
  - Number of CPUs allocated for CPU licenses
- Database information:
  - Metadata database and ODBC driver
  - Data warehouse database and ODBC driver

## Is the information sent to MicroStrategy secure?

Yes. The Activation XML file is sent to MicroStrategy in the following ways:

- Automatically during installation or through License Manager. This information is encrypted using a RIPEMD-160 algorithm before being sent to MicroStrategy.
- Manually in the <https://licensing.microstrategy.com> website. Communication with this website is conducted through Secure Socket Layer once the user has been authenticated.

## Can I change the information in the Activation XML file?

The content of this file is secured with a digital signature. Contact MicroStrategy Technical Support if the content is incorrect.

## What is used to lock the server to a machine?

A unique hardware identifier for each machine is used to lock an installation to that machine. Any changes to these identifiers will require reactivation of the Server Installation:

- Windows: An identifier generated from a one-way hash of the network interface card MAC address
- Linux: An identifier generated from a one-way hash of the network interface card MAC address

## Is there a grace period from the time server products are installed to when it can be activated?

Yes. There is a 30 calendar day grace period from installation (7 days for Evaluation Edition) before a server installation must be activated.

## **What happens if the Server Installations are not activated?**

If a server installation has not been activated within 30 calendar days (7 days for Evaluation Edition), the server products will not be able to be restarted.

## **Should the installations be automatically or manually activated and deactivated?**

It is preferable to automatically send the activation information to MicroStrategy, either during installation or using License Manager. This is an easy process that should take less than one minute to complete, and ensures that the correct information is sent to MicroStrategy.

## **What should be done if requesting an Activation Code fails during installation?**

Firstly, allow the installation to complete. The Activation Code request should then be attempted using License Manager. If this does not work, activate the Server Installation manually by visiting the Activation website at <https://licensing.microstrategy.com>.

## **What should be done if automatic request for an Activation Code does not work at all?**

The Server Installation should be manually activated using License Manager on a machine that has access to the Web. The Activation XML file should be copied from the Server Installation that requires activation to this computer. If this does not work, contact MicroStrategy Technical Support.

## **Can the Activation Code be used on a different machine?**

No. The Activation Code contains the unique identifier for a specific machine and can only be used on that machine. The server products will not be activated if the incorrect Activation Code is used.

## **What if the Server installation has to be moved to another machine?**

If the Server Installation needs to be moved to another machine, uninstall the MicroStrategy products or deactivate the Server Installation from License Manager. Notify MicroStrategy that Server Installation has been deactivated. This keeps your active server inventory up to date as shown to you on <https://licensing.microstrategy.com>.

## **What if the server machine has a catastrophic failure and cannot be deactivated automatically or manually?**

If a machine has a catastrophic failure and the server products cannot be uninstalled or deactivated, contact MicroStrategy Technical Support to update the status of this Server.

## **If there is more than one server product installed on a machine and one is removed, does the Server Installation need to be deactivated?**

Upon removal of a server product, all remaining server products are automatically deactivated. The remaining server product(s) need to be reactivated to reflect the new product configuration on that machine. For example, if a machine contains MicroStrategy Narrowcast Server and MicroStrategy Web, and MicroStrategy Narrowcast Server is removed, MicroStrategy Web will be automatically deactivated. MicroStrategy Web will need to be reactivated.

## **What information can be monitored on the website?**

When registered Technical Support liaisons log into the MicroStrategy Activation website, they can display a list of all Server Installations. This report lists each Server Installation along with the following information for reference:

- Installation ID
- Activation Status
- Contract ID
- Operating System
- Machine CPUs
- Last Update Type
- Last Updated by
- Last Update Date
- Products installed
- Activation and deactivation history

The Installation ID is a unique identifier for Server Installations. This ID is provided along with the Activation Code in the email received when requesting activation.

## **Does Server Activation apply to MicroStrategy Suite?**

Yes. The MicroStrategy Suite must be activated within 30 days of installation.

# CONFIGURING AND CONNECTING INTELLIGENCE SERVER

After installing MicroStrategy, you must complete a few configuration tasks. This chapter addresses the processes used to configure the databases you intend to use in your business intelligence system, as well as an installed MicroStrategy suite of products.



The MicroStrategy platform includes a Tutorial project, which is a sample data warehouse and a demonstration project you can use to learn about the various features that MicroStrategy offers. It is ready to be used and requires no additional configuration tasks. If you want to set up a new system using your own data, you must read this chapter and perform the tasks it describes.

This chapter includes the following information:

If you are configuring MicroStrategy on a Linux machine that does not have a GUI, you can perform configuration tasks with command line tools. For steps to perform configuration tasks using command line tools in Linux, see [Chapter 12, Configuring MicroStrategy Using Command Line Tools](#).

## Communicating with databases

Establishing communication between MicroStrategy and your databases or other data sources is an essential first step in configuring MicroStrategy products for reporting and analyzing data. This section explains how MicroStrategy communicates with various data sources and the steps required to set up this communication.

ODBC (Open Database Connectivity) is a standard database access method. ODBC enables a single application to access database data, regardless of the database management system (DBMS) that stores the data. A DBMS is a collection of programs that enables you to store, modify, and extract information from a database.

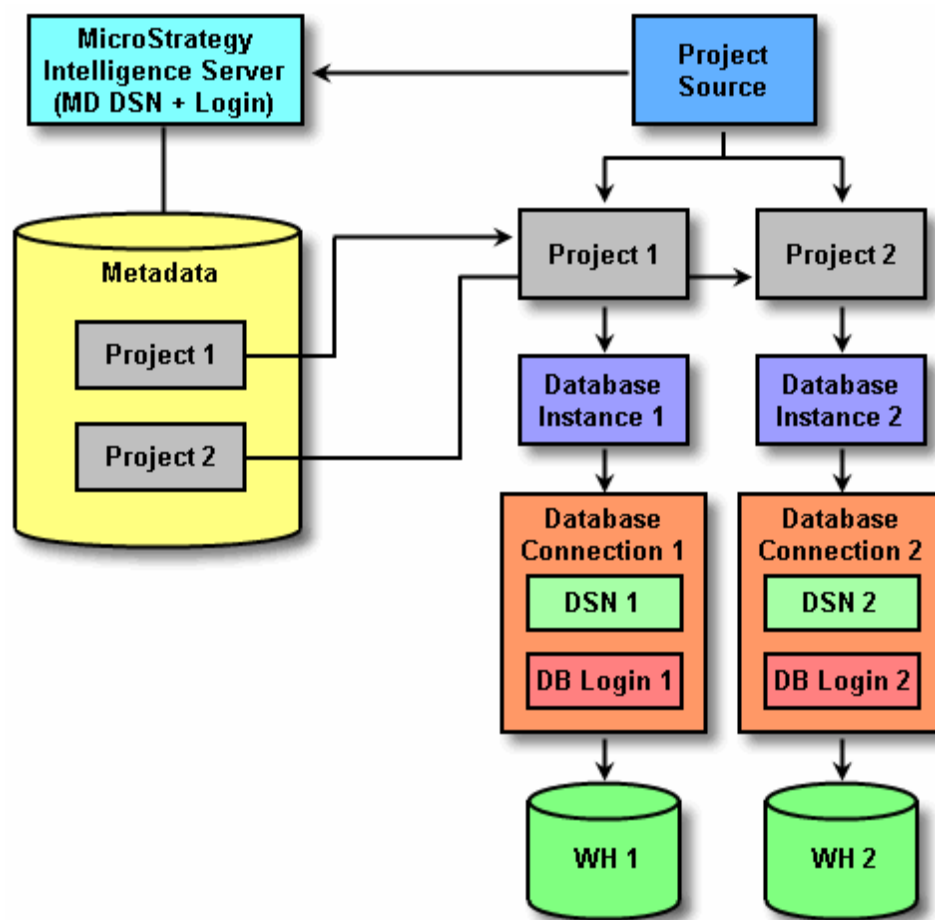
MicroStrategy Intelligence Server, when used in a three- or four-tier configuration, is the application that uses ODBC to access a DBMS. ODBC drivers translate MicroStrategy Intelligence Server requests into commands that the DBMS understands. MicroStrategy

Intelligence Server connects to several databases (at a minimum, the data warehouse and the metadata repository) to do its work.

Users of MicroStrategy Web can also connect to data sources using database connections. A database connection supports connecting to data sources through the use of DSNs, as well as through DSNless connections, to import and integrate data into MicroStrategy. For steps to create database connections in MicroStrategy Web, see [Creating database connections in Web, page 426](#).

This section describes the ODBC standard for connecting to databases and creating data source names (DSNs) for the ODBC drivers that are bundled with the MicroStrategy applications.

The diagram below illustrates the three-tier metadata and data warehouse connectivity used in the MicroStrategy system.



The diagram shown above illustrates projects that connect to only one data source. However, MicroStrategy allows connection to multiple data sources in the following ways:

- With MicroStrategy MultiSource Option, a MicroStrategy project can connect to multiple relational data sources. For information on MultiSource Option, see the [Project Design Guide](#).

- You can integrate MDX cube sources such as SAP BW, Microsoft Analysis Services, and Hyperion Essbase with your MicroStrategy projects. For information on integrating these MDX cubes sources into MicroStrategy, see the [MDX Cube Reporting Guide](#).

This section provides information and instructions on the following tasks:

## Setting up ODBC

The following information assists you in setting up ODBC between Intelligence Server and your metadata database and data warehouse.

ODBC is a standard method of communicating with database servers. Intelligence Server uses ODBC to connect to and communicate with all database servers in the system. Specifically, ODBC connects to and transfers data to and from data sources within relational databases.

ODBC permits maximum interoperability—an application can access data in diverse DBMSs through a single framework. A client application uses a database driver and a driver manager to make a connection to the data source. A data source, identified by a data source name, is the database or file accessed by a driver. Data source is another term for a logical database within a database server. A database server can contain multiple logical databases or data sources.

When setting up your MicroStrategy environment, you must create a separate connection to the data warehouse and metadata repository. This requirement is true even if both databases are accessed through the same DBMS. Further description of these two requirements is below:

- A data warehouse stores the data that users of the system must analyze to track and respond to business trends, and to facilitate forecasting and planning efforts.
- Metadata is a repository whose data associates the tables and columns of a data warehouse with user-defined attributes and facts to enable the mapping of business views, terms, and needs to the underlying database structure. Metadata can reside on the same server as the data warehouse or on a different server. It can be stored in different relational DBMSs.

A successful ODBC connection requires the following information:


- A data source name (DSN) is the name for a pointer used by a client application to find and connect to a data source. A data source is the database accessed by a driver. The information obtained through a DSN generally includes the host computer name or IP address, instance name, and database name. However, the exact information varies depending on the type of database server.
- An ODBC driver is a type of software that translates information between the client application (Intelligence Server) and the database server API. For more information on ODBC drivers and how they work with MicroStrategy, see [ODBC drivers, page 160](#).
- 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 203](#)).

## 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 and Linux that are certified for Intelligence Server and different DBMS types, see [Certified ODBC drivers for MicroStrategy Intelligence Server, page 72](#).

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 Linux: From the 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.
- 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 Linux

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

The ODBC driver manager and support libraries are usually installed in `INSTALL_PATH/lib`



The database-specific ODBC drivers are installed in the locations specified during the installation of the drivers. MicroStrategy-branded ODBC drivers are installed in `INSTALL_PATH/lib`, where `INSTALL_PATH` is the directory you specified as the Install Directory in the Install Wizard.



The MicroStrategy Connectivity Wizard lists only the MicroStrategy-branded ODBC drivers. However, this guide also provides information on how to install drivers from other vendors with MicroStrategy. For more information, see [Creating DSNs for specific data sources, page 392](#).

## 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 161](#)).
- 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 163](#)).



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 Linux using the Connectivity Wizard interface, perform the following steps:
  - In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.
  - Browse to the folder `bin` and type `./mstrconnectwiz`, and then press ENTER. The Welcome page of the Connectivity Wizard opens.
- On Linux from the command line, then perform the following steps:
  - In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.
  - Browse to the folder `bin` and type `./mstrconnectwiz -h`, and then press ENTER.

This command displays command line operation syntax and examples for different database platforms. Create your command based on the syntax and examples displayed. Once you perform your command, the DSN is created and 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 361](#) in *Chapter 12, 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 Linux. For a list of the available ODBC drivers for Windows and Linux that are certified for Intelligence Server and different DBMS types, see [Certified ODBC drivers for MicroStrategy Intelligence Server, page 72](#).
- 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 392](#).
- 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.

- 4 Click **Add**. The Create New Data Source dialog box opens.
- 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 392](#).

- 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.

| Layer             | Test with                                                           |
|-------------------|---------------------------------------------------------------------|
| ODBC driver       | Test ODBC<br><br>mstrtestodbc or mstrtodbcdx                        |
| Network<br>TCP/IP | Simple Network Layer Testing Tool<br><br>Ping, PING.EXE, for TCP/IP |



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 12, Configuring MicroStrategy Using Command Line Tools*.
  - On Linux using the DB Query Tool interface, perform the following steps:
    - In a Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory that you specified as the home directory during installation.
    - Browse to the folder `bin` and type `./mstrdbquerytool`, then press `ENTER`.
  - On Linux from the command line, perform the following steps:
    - In a Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory that you specified as the home directory during installation.
    - Browse to the folder `bin` and type `./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 12, Configuring MicroStrategy Using Command Line Tools*.
- 2** From the **Session** menu, select **Open Connection**, or click the **Connect** icon on the toolbar. The Connect dialog box opens. The connection interface varies depending on the destination database.
  - 3** Select the DSN for a data source.
  - 4** Enter the appropriate user name and password.
  - 5** Click **Connect**. After your connection is opened, the connection string is displayed in the MicroStrategy DB Query Tool at the bottom. Your cursor is inserted automatically in the SQL Statement window.
  - 6** In the SQL Statement window, type a SQL query such as:
 

```
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 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 188](#).

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 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 157](#)). For a list of certified metadata platforms, see the *MicroStrategy Readme*.
- In a Linux environment, the Configuration Wizard must be able to communicate with Intelligence Server over TCP/IP network protocol. To achieve this, the `hosts` file in the `/etc` directory must include one entry identifying the local host in the form:

***IP-address local-machine-name***

For example, `123.4.5.6 FakeISmachine`. Modifying the `hosts` file may require an account with root privileges.

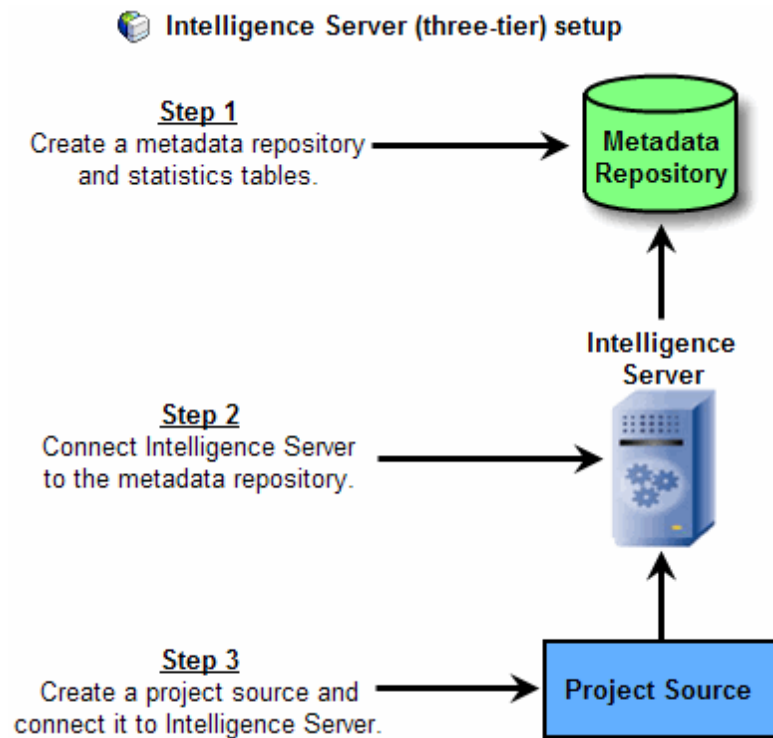
- MicroStrategy products must be configured on the machine on which they are installed. You cannot configure them remotely.

## Configuring MicroStrategy software

The MicroStrategy Configuration Wizard opens automatically after you install MicroStrategy products and restart your machine.

You can configure a MicroStrategy Web and Intelligence Server (four-tier), Intelligence Server (three-tier), or direct (two-tier) setup for MicroStrategy.

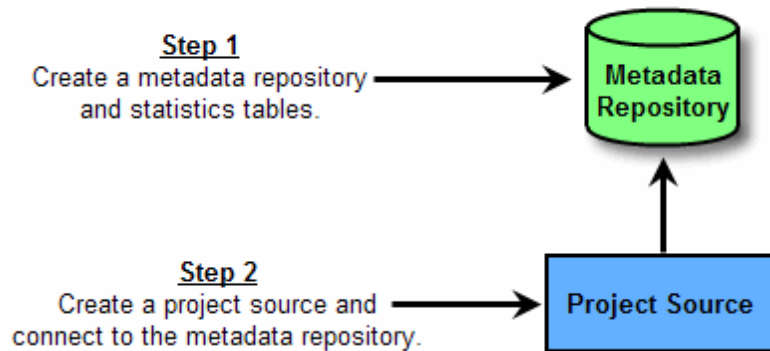
The following figure describes how to configure MicroStrategy to suit an Intelligence Server (three-tier) environment. It also shows how the various components of the MicroStrategy Configuration Wizard, the metadata repository, Intelligence Server, and the project source interact with each other.



A MicroStrategy Web (four-tier) setup involves configuring a web server to communicate between Intelligence Server and MicroStrategy Web. For more information on deploying MicroStrategy Web, see [Chapter 7, Deploying MicroStrategy Web and Mobile Server](#).

The figure below describes how to configure MicroStrategy to suit a direct (two-tier) environment. It also shows how the various components of the MicroStrategy Configuration Wizard, the metadata repository and the project source interact with each other.

### Direct (two-tier) setup



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 169](#).
- 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 188](#).

- Linux using the Configuration Wizard interface, then perform the following steps:
  - From a Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory that you specified as the home directory during installation.
  - Browse to the folder `bin` and type `./mstrcfgwiz`, then press **ENTER**. The Configuration Wizard opens. Continue to the steps provided in [To select a configuration task, page 169](#).



- Linux from the command line, then perform the following steps:
  - From a Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory that you specified as the home directory during installation.
  - Browse to the folder `bin` and type `./mstrcfgwiz-editor`, then 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 349](#).

## To select a configuration task

- 2 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 170](#).
  - **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 179](#).
  - **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 184](#).
  - **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 171](#)
- [Creating a History List repository, page 174](#)
- [Creating statistics and Enterprise Manager repositories to maintain and monitor system activity, page 176](#)

As you complete the configuration process, messages may be displayed. For details on system messages displayed during the configuration process, see [Configuration messages, page 178](#).

You can choose to create metadata, History List, and statistics repositories using a response file with the Configuration Wizard. This lets you provide users with a configuration file to complete the configuration tasks rather than requiring users to step through the Configuration Wizard. Creating and using a response file can be done using the Configuration Wizard interface or a command line tool available for Linux. The steps to perform these two configuration options are provided in the sections listed below:

- [Configuring MicroStrategy with a response file, page 188](#)
- [Configuring MicroStrategy with a response.ini file, page 364](#) in *Chapter 12, Configuring MicroStrategy Using Command Line Tools*

## 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:

| Database Object | Type of Permissions Required                 |
|-----------------|----------------------------------------------|
| Tables          | Select, Insert, Update, Create, Drop, Delete |
| Indexes         | Create, Drop                                 |
| Triggers        | Create, Drop                                 |
| Functions       | Create, Execute                              |
| Packages        | Create                                       |
| Procedures      | Create, Execute                              |



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 170](#) 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 444](#) 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 168](#).
- 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 161](#).



Although it is possible to use the Microsoft Access database for the metadata repository, it is not a suitable metadata repository for a production project. You should not use Microsoft Access for anything other than a proof-of-concept or demonstration type of application.

- 5 Type a **User Name** and **Password** that can connect to the data source.

The database user that you provide becomes the owner of all metadata tables and objects. The database user is required to have the Select, Insert, and Update permissions. Intermediate tables are created in the metadata for recursive search queries, which requires Create and Drop permissions as well. Updating the schema requires the Delete permission.

- 6 After providing a valid user name and password, you can click **SQL Preview** to open the SQL Preview dialog box. This dialog box provides the SQL statements that will be executed on your data source to create the metadata tables. Click **Close** 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

- 7 Click **Advanced**. Options to specify a table prefix and a SQL script to create metadata tables are displayed.
- 8 In the **Table Prefix** field, you can specify a prefix to be used when metadata tables are created in the database you select. This is an optional configuration. However, you must use different prefixes for your metadata tables and your History List tables if you store them in the same database.

Most databases use a prefix of two characters. However, you can supply as many letters, numbers, underscores ( \_ ), and periods ( . ) as required to support your database prefixes. To determine character limits for a prefix, refer to your third-party database vendor documentation.

- 9 In the **Script** field, a SQL script to create metadata tables optimized for your database is selected. If you want to select a different script, click ... to browse to and select a customized script. For more information on the default SQL scripts, see [SQL scripts, page 178](#).
- 10 Click **Next**. The next configuration page that opens depends on your configuration scenario:
  - If your metadata repository does not need to be upgraded and you did not select to create History List or statistics tables, the Summary page opens. You can complete your configuration as described in [To review and save your metadata configuration, page 173](#) 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 174](#) and [Creating statistics and Enterprise Manager repositories to maintain and monitor system activity, page 176](#).
  - 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

- 11 Review the summary information.

You can click **Save** to save the configuration as a response ( .ini ) file to configure metadata repositories on other systems or to run silent configurations at a later time. For information on running the Configuration Wizard with a response file, see [Configuring MicroStrategy with a response file, page 188](#).

- 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 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 203](#).



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 170](#) 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 171](#).
- 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 444](#) 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 168](#).
- 2 Select **Metadata, History List and Statistics Repository Tables** and click **Next**. The Repository Configuration: Repository Types page opens.
- 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 161](#).

- 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 203](#).

- 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 178](#).

- 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 176](#) 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 176](#).

## 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 188](#).

### 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 203](#).

## 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 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 203](#).



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 170](#) 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 171](#) or [Creating a History List repository, page 174](#), 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 444](#) 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 168](#).
- 2 Select **Metadata, History List and Statistics Repository Tables** and click **Next**. The Repository Configuration: Repository Types page opens.
- 3 Select the **Statistics & Enterprise Manager Repository** check box and click **Next**. The Repository Configuration: Statistics & Enterprise Manager Repository page opens.
- 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 161](#).

- 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 178](#).
- 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 188](#).

- 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 203](#).

## 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 179](#) 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 Linux. The steps to perform these two configuration options are provided in the sections listed below:

- [Configuring MicroStrategy with a response file, page 188](#)
- [Configuring MicroStrategy with a response.ini file, page 364 in Chapter 12, Configuring MicroStrategy Using Command Line Tools](#)

### 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 168](#).
- 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 161](#))
- **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. For details about passwords and other user management information, see the [System Administration Guide](#)

- 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 about the Intelligence Server Configuration Editor, see 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 181](#) 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 181](#) 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 183](#) 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 443](#).
  - **Register Intelligence Server as a Service:** This option is only available if you are configuring Intelligence Server on a Linux machine, and you have root access and permissions to the Linux machine that Intelligence Server is installed on.

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 Linux, you must stop the service, and then use the `mstrectl` command line tool. The syntax is `mstrectl -s IntelligenceServer us`, where `IntelligenceServer` is the name of a server definition. For information about starting, stopping, and registering Intelligence Server as a service, see the [System Administration Guide](#).

- **Projects to load at startup:** This pane displays all the projects that are in the metadata repository. You can select projects to use with the server definition that you have chosen. The projects that you select are loaded on the server at startup.
- **Start Intelligence Server when finished:** Select this check box to have Intelligence Server start once you complete this configuration.

If you use Windows NT authentication with SQL Server, you must type the Windows NT account user name and password in the Service Manager to successfully start

Intelligence Server. For information on how to access and use the Service Manager, see the [System Administration Guide](#).

- **Identify missing DSNs:** Select this check box to verify that all DSNs, which are used for database instances created in MicroStrategy, are locally available. This helps to ensure that your database instances in MicroStrategy can connect successfully to their associated data sources.

By default, this check box is cleared, which means the availability of all local DSNs used in database instances is not verified. While this may mean that all DSNs used in database instances are not available, it can save system resources required for the Intelligence Server configuration process.

**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 communicate using 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](#).

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](#).

**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 188](#).

**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, see the [System Administration Guide](#).

## Creating project sources

Project sources represent a connection to a metadata database or a MicroStrategy Intelligence Server. The project source stores the location of the metadata repository or the MicroStrategy Intelligence Server definition that is used to run the project. Through a project source you can create, manipulate, and administer MicroStrategy projects.

When you create a metadata repository, by default it creates a server (three-tier) project source. You can use the **Project Sources** option in the MicroStrategy Configuration Wizard if you need to create multiple project sources or a direct (two-tier) project source. The steps to create the different types of project sources are:

- [Creating a direct \(two-tier\) project source, page 184](#): Direct project sources that connect directly to the metadata through ODBC. You cannot create a direct project source on Linux.
- [Creating a server \(three-tier\) project source, page 185](#): Server project sources that connect to the metadata through an Intelligence Server.

You can choose to create project sources using a response file with the Configuration Wizard. This lets you provide users with a configuration file to complete the configuration tasks rather than requiring users to step through the Configuration Wizard. Creating and using a response file can be done using the Configuration Wizard interface or a command line tool available for Linux. The steps to perform these two configuration options are provided in the sections listed below:

- [Configuring MicroStrategy with a response file, page 188](#)
- [Configuring MicroStrategy with a response.ini file, page 364](#) in *Chapter 12, Configuring MicroStrategy Using Command Line Tools*

## Creating a direct (two-tier) project source

A direct project source is used to connect directly to the metadata repository using ODBC. A direct project source connection does not allow you to access MicroStrategy Web, run Report Services documents, or use any of the other MicroStrategy features that are provided through Intelligence Server.

You cannot create a direct project source on Linux.

## 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 168](#).
- 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 161](#)) 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 187](#).
- 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 188](#).

- 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 168](#).
- 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.
- 8 Click **Next**. The Project Source Creation: Authentication page opens.
- 9 Select the authentication mode for the project source. For information on the available authentication modes, see the [Authentication modes, page 187](#).
- 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 188](#).

- 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 187*
- *Login ID and password entered by the user: Standard authentication, page 187*
- *Guest account: Anonymous authentication, page 187*
- *LDAP authentication, page 188*
- *Login ID and password entered by the user for the warehouse: Database authentication, page 188*
- *Integrated authentication, page 188*

For information on the benefits of the various authentication modes and other authentication topics, see the [System Administration Guide](#)

### Network login ID: Windows authentication

To use Windows authentication, you must create users in the MicroStrategy environment and then link them to Windows users. If you use Windows as your network operation system and your users are already defined in the Windows directory, your users can access the MicroStrategy application without having to enter a login ID and password.

### Login ID and password entered by the user: Standard authentication

When using standard authentication, the MicroStrategy Intelligence Server is the authentication authority. Intelligence Server verifies and accepts the login and password provided by the user. This information is stored in the metadata repository. When a project source is configured to use standard authentication, users must enter a valid login ID and password combination before they can access the project source. Each user has a unique login/password and can be identified in the MicroStrategy application uniquely. By default, all users connect to the data warehouse using one RDBMS login ID, although you can change this using connection mapping. For information on configuring connection mapping, see the [System Administration Guide](#).

### Guest account: Anonymous authentication

When using anonymous authentication, users log in as Guest and do not need to provide a password. By default, guest users can access the project, browse objects, run and manipulate reports, but they cannot create their own objects or schedule report executions. However, you determine what the Guest user can and cannot do by modifying the Public

user group. Guest users inherit their privileges from the Public group; they are not part of the Everyone group.

### **LDAP authentication**

Lightweight Directory Access Protocol (LDAP) authentication identifies users within a repository of users stored in an LDAP server (such as Novell Directory Services). If you use an LDAP directory to centrally manage users in your environment, you may want to use LDAP authentication. Group membership can be maintained in the LDAP directory without having to also be defined in the MicroStrategy Intelligence Server. When using LDAP authentication, LDAP users or groups are linked to users or groups in the MicroStrategy environment.

### **Login ID and password entered by the user for the warehouse: Database authentication**

This mode of database authentication identifies users using a login ID and password stored in the data warehouse database. Under this mode of authentication, a warehouse database is associated with each project. When users log in to a project source, they are logging in to the Intelligence Server. Use database authentication if you want the data warehouse RDBMS to be the authority for identifying users and you do not want to maintain user credentials in the Intelligence Server as well as the RDBMS.

### **Integrated authentication**

Integrated authentication enables a Windows user to log in once to their Windows machine. The user does not need to log in again separately to MicroStrategy Developer or MicroStrategy Web. This type of authentication uses Kerberos to validate a user's credentials.

## **Configuring MicroStrategy with a response file**

The Configuration Wizard walks you through the process of setting up the environment for the MicroStrategy products installed in your system. You can also configure server definitions, project source names, an Enterprise Manager project, and the metadata, History List, and statistics repositories using a response file with the Configuration Wizard. This enables you to provide a configuration file to users to complete the configuration tasks, rather than requiring users to step through the Configuration Wizard. This can be done to configure a MicroStrategy installation on Windows and Linux.

The Configuration Wizard can also be used to perform MicroStrategy upgrades, 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 189](#) in this section. This ensures that all applicable options are included in the response file with valid values.

However, you can also create and use a response file with the Configuration Wizard in command line mode on Linux machines. For steps to create and use a response file as well as perform other configurations using command line tools in Linux, see the [Configuring MicroStrategy with a response.ini file, page 364](#) section in [Chapter 12, Configuring MicroStrategy Using Command Line Tools](#).

---

## To create a response file

---

- 1 Open the MicroStrategy Configuration Wizard. To do this, see [To configure MicroStrategy through the Configuration Wizard, page 168](#).
- 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 170](#)
  - [Setting up MicroStrategy Intelligence Server, page 179](#)
  - [Creating project sources, page 184](#)
- 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 190](#) 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 191](#).

## 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 190](#): 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 190](#): This covers the procedure of running a response file from the Windows command line. This enables users to run the file without using any graphical user interfaces.

If you are configuring a MicroStrategy installation on Linux, you can use a command line version of the Configuration Wizard to create and use a response file. For steps to create and use a response file as well as perform other configurations using command line tools in Linux, see the [Configuring MicroStrategy with a response.ini file, page 364](#) section in [Chapter 12, Configuring MicroStrategy Using Command Line Tools](#).

- **Configuring MicroStrategy components with System Manager:** You can use a Configuration Wizard response file as part of an System Manager workflow. System Manager lets you define multiple configurations for your MicroStrategy environment that can be executed in a single workflow. For information on using MicroStrategy System Manager to configure and deploy your MicroStrategy environments, see the [System Administration Guide](#).

---

### To use a response file with the Configuration Wizard

---

- 1 Open the MicroStrategy Configuration Wizard. To do this, see [To configure MicroStrategy through the Configuration Wizard, page 168](#).
- 2 Click **Load**. The Open dialog box displays.
- 3 Browse to the path where the response file is saved and click **Open**. The Summary page opens.
- 4 An overview of all of the configuration tasks performed by the response file is displayed. Review the configuration tasks and click **Finish** to perform the configuration. The summary information is updated as the configurations are completed, providing a way to track the progress of the configurations.

---

### To use a response file through the Windows command line

---

The steps below are specific to configuring MicroStrategy installed on Windows. For steps to create and use a response file as well as perform other configurations using command line tools in Linux, see [Chapter 12, Configuring MicroStrategy Using Command Line Tools](#).

- 1 Type the following command in the Windows command line:

```
macfgwiz.exe -r "Path\response.ini"
```

Where *Path\* is the fully qualified path to the response file. For example, the common location of a response file is:

```
C:\Program Files\Common Files\MicroStrategy\RESPONSE.INI
```

- 2 If an error message is displayed, check the path and name you supplied for the response file and make any required changes. Repeat the previous step to execute the configuration.

## Response configuration parameters and options

It is recommended that you always create the response file through the GUI mode of the Configuration Wizard. However, you can also modify a response file with a text editor to make minor changes such as entering different user login and password information.



The file must be saved with ANSI encoding.

The response file for configuring MicroStrategy is divided into three areas of configuration, which are described in the sections below:

- [Creating metadata, History List, and statistics repositories, page 191](#)
- [Setting up MicroStrategy Intelligence Server, page 193](#)
- [Creating and configuring project sources, page 197](#)

## 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.

| Options         | Description                                                                                                                                                                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [Repository]    | 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.                                                                                                                          |
| Repository =    | Defines whether a metadata, History List, and statistics repositories are configured, as determined by the following values: <ul style="list-style-type: none"> <li>• 1 : Configures metadata, History List, and statistics repositories.</li> <li>• 0 : Does not configure metadata, History List, and statistics repositories.</li> </ul> |
| CreateMDTables= | Defines whether metadata tables are created in a metadata repository, as described below: <ul style="list-style-type: none"> <li>• 1 : Creates metadata tables in the metadata repository and creates a default configuration</li> <li>• 0 : Does not create metadata tables in a metadata repository</li> </ul>                            |

| Options                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CreateHistListTables=<br>=   | Defines whether a History List repository is created, as determined by the following values: <ul style="list-style-type: none"> <li>1 : Creates a History List repository.</li> <li>0 : Does not create a History List repository.</li> </ul>                                                                                                                                                                                                |
| CreateStatisticsTables=<br>= | Defines whether statistics and Enterprise Manager repositories are created, as determined by the following values: <ul style="list-style-type: none"> <li>1 : Creates statistics and Enterprise Manager repositories.</li> <li>0 : Does not create statistics and Enterprise Manager repositories.</li> </ul>                                                                                                                                |
| MetadataPath=<br>=           | Locates the SQL scripts for creating the metadata tables. Example paths to SQL scripts in different environments are listed below: <ul style="list-style-type: none"> <li>64-bit Windows environment: C:\Program Files (x86)\Common Files\MicroStrategy\MD8SQL8.sql.</li> <li>Linux: /INTELLIGENCE_SERVER_INSTALL_PATH/mdsql.sql.</li> </ul>                                                                                                 |
| HistoryListPath=<br>=        | Locates the SQL scripts for creating the History List repository. Example paths to SQL scripts in different environments are listed below: <ul style="list-style-type: none"> <li>64-bit Windows environment: C:\Program Files (x86)\Common Files\MicroStrategy\content_server_db_Oracle.sql.</li> <li>Linux: /INTELLIGENCE_SERVER_INSTALL_PATH/content_server_db_Oracle.sql.</li> </ul>                                                     |
| StatisticsPath=<br>=         | Locates the SQL scripts for creating the statistics and Enterprise Manager repositories. Example paths to SQL scripts in different environments are listed below: <ul style="list-style-type: none"> <li>64-bit Windows environment: C:\Program Files (x86)\Common Files\MicroStrategy\StatisticsEnterpriseManagerScripts\DDLScripts\CreateTablesScript.sql</li> <li>Linux: /INTELLIGENCE_SERVER_INSTALL_PATH/statistics_DB2.sql.</li> </ul> |
| DSNName=<br>=                | Defines the Data Source Name for configuring a metadata repository in the ODBC database.                                                                                                                                                                                                                                                                                                                                                     |
| UserName=<br>=               | Defines the user name to log in to the database containing the metadata repository.                                                                                                                                                                                                                                                                                                                                                          |
| UserPwd=<br>=                | Defines the password to log in to the database containing the metadata repository.                                                                                                                                                                                                                                                                                                                                                           |
| DSNNameHist=<br>=            | Defines the Data Source Name for configuring the History List repository in the ODBC database.                                                                                                                                                                                                                                                                                                                                               |
| UserNameHist=<br>=           | Defines the user name to log in to the database for configuring the History List                                                                                                                                                                                                                                                                                                                                                             |



| Options          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| st=              | repository.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| UserPwdHist=     | Defines the password to log in to the database for configuring the History List repository.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| DSNNameStats=    | Defines the Data Source Name for configuring the statistics and Enterprise Manager repositories in the ODBC database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| UserNameStats=   | Defines the user name to log in to the database for configuring the statistics and Enterprise Manager repositories.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| UserPwdStats=    | Defines the password to log in to the database for configuring the statistics and Enterprise Manager repositories.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| EncryptPassword= | <p>Defines whether the password is encrypted in the response file, as determined by the following values:</p> <ul style="list-style-type: none"> <li>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.</li> <li>1 : Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</li> </ul> |
| DBName=          | 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.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| TBName=          | Defines the tablespace name to be created in the database. This option should only be used when connecting to a DB2 z/OS database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

## 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.

| Options               | Description                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>[Server]</code> | In this section you can configure the Intelligence Server definition. You can have more than one <code>[Server]</code> section. Additional server sections can be included as <code>[Server1]</code> , <code>[Server2]</code> , and so on. |

| Options               | Description                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Server=               | <p>Defines whether MicroStrategy Intelligence Server is configured, as determined by the following values:</p> <ul style="list-style-type: none"> <li>• 1 : Configures MicroStrategy Intelligence Server</li> <li>• 0 : Does not configure MicroStrategy Intelligence Server</li> </ul>                                                                                                  |
| Action=               | <p>Defines whether a server definition is created, used, or deleted, as determined by the following values:</p> <ul style="list-style-type: none"> <li>• 1 : Creates a new server definition</li> <li>• 2 : Uses an existing server definition</li> <li>• 3 : Deletes an existing server definition</li> <li>• 4 : Creates a new server definition and uses it as the default</li> </ul> |
| InstanceName=         | <p>Defines the name of the Intelligence Server instance.</p> <p>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, <code>InstanceName=ServerInstance1\ServerInstance2</code>.</p>                                                                      |
| ProjectsToRegister=   | <p>Defines projects to be loaded when Intelligence Server is started. You can select to load multiple projects, separating projects by the \ character. For example, <code>ProjectsToRegister=Project1\Project2</code>.</p>                                                                                                                                                              |
| ProjectsToUnRegister= | <p>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, <code>ProjectsToUnRegister=Project1\Project2</code>.</p>                                                                                                                                                    |
| DSName=               | <p>Defines the data source name for configuring the MicroStrategy Intelligence Server. This is the data source that stores the metadata.</p>                                                                                                                                                                                                                                             |
| DSNUser=              | <p>Defines the user name to log in to the metadata database.</p>                                                                                                                                                                                                                                                                                                                         |
| DSNPwd=               | <p>Defines the password to log in to the metadata database.</p>                                                                                                                                                                                                                                                                                                                          |

| Options                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EncryptPassword=         | <p>Defines whether the password is encrypted in the response file, as determined by the following values:</p> <ul style="list-style-type: none"> <li>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.</li> <li>1 : Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</li> </ul> |
| DSSUser=                 | Defines the MicroStrategy user name to log in to the project.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| DSSPwd=                  | Defines the password for the MicroStrategy user name to log in to the project.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| MDPrefix=                | Defines a prefix for metadata repository tables used by the server definition.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| UseAsDefault=            | <p>Defines whether the Intelligence Server definition is set as the default server definition to use for Intelligence Server, as determined by the following values:</p> <ul style="list-style-type: none"> <li>True : Defines the Intelligence Server definition as the default server definition</li> <li>False: Does not define the Intelligence Server definition as the default server definition</li> </ul>                                                                                                                                                                                                                                                                                               |
| Port=                    | Defines the port used by the Intelligence Server.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| RegisterAsService=       | <p>This option is only available on Intelligence Servers running on Linux operating systems.</p> <p>Defines whether Intelligence Server is registered as a service. Registering Intelligence Server as a service is determined by the following values:</p> <ul style="list-style-type: none"> <li>1 : Registers Intelligence Server as a service. Performing this task requires a Linux login with root level access and privileges.</li> <li>0 : Does not register Intelligence Server as a service.</li> </ul>                                                                                                                                                                                               |
| StartServerAfter Config= | <p>Defines whether Intelligence Server is started after applying the configuration, as determined by the following values:</p> <ul style="list-style-type: none"> <li>1 : Intelligence Server is started after successfully applying the configuration.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Options                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                 | <ul style="list-style-type: none"> <li>0 : Intelligence Server is not started after applying the configuration.</li> </ul>                                                                                                                                                                                                                                                                                                                            |
| ConfigureSSL=                   | <p>Defines whether to enable Intelligence Server and Developer to communicate using the SSL protocol, as determined by the following values:</p> <ul style="list-style-type: none"> <li>1 : Enables the use of the SSL protocol for Intelligence Server and Developer communications.</li> <li>0 : Disables the use of the SSL protocol for Intelligence Server and Developer communications.</li> </ul>                                              |
| SSLPort=                        | Defines the port to use for SSL access. By default, the port is 39321.                                                                                                                                                                                                                                                                                                                                                                                |
| CertificatePath=                | Locates the SSL certificate file you created for Intelligence Server. Type the full path to the SSL certificate file.                                                                                                                                                                                                                                                                                                                                 |
| KeyPath=                        | Locates private key file you created while requesting the certificate for Intelligence Server. Type the full path to the private key file.                                                                                                                                                                                                                                                                                                            |
| KeyPassword=                    | Defines the password that you used while creating the private key for the SSL certificate.                                                                                                                                                                                                                                                                                                                                                            |
| DefaultStatisticsRep            | <p>Specifies whether you can create a default statistics database instance for the all of the projects of the local Intelligence Server metadata, as determined by the following values:</p> <ul style="list-style-type: none"> <li>1: You can create a default statistics database instance, using the statistics parameters listed in this table below.</li> <li>0: A default statistics database instance is not created.</li> </ul>               |
| DefaultDSNNameDefaultStatistics | Specifies the data source name for your statistics repository.                                                                                                                                                                                                                                                                                                                                                                                        |
| UserNameDefaultStatistics       | Specifies the database user name for the user that can connect to the statistics data source.                                                                                                                                                                                                                                                                                                                                                         |
| UserPwdDefaultStatistics        | Specifies the password for the user that can connect to the statistics data source.                                                                                                                                                                                                                                                                                                                                                                   |
| EncryptUserPwdDefaultStatistics | <p>Defines whether the statistics user password is encrypted in the response file, as determined by the following values:</p> <ul style="list-style-type: none"> <li>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</li> </ul> |

| Options                     | Description                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             | <p>aware that this can compromise your database security if you do not remove the password from the response file before distributing it.</p> <ul style="list-style-type: none"> <li>1 : Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</li> </ul>                                                                        |
| DefaultStatisticsPrefix     | Defines a prefix for statistics repository tables used by the server definition.                                                                                                                                                                                                                                                                                                                 |
| DefaultStatisticsBasicStats | <p>Defines whether to enable basic statistics, as determined by the following values:</p> <ul style="list-style-type: none"> <li>1: Enables the logging of basic statistics for new projects and any projects that are not now logging statistics.</li> <li>0: Does not enable the logging of basic statistics for new projects and any projects that are not now logging statistics.</li> </ul> |

## 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.

| Options                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>[Client]</code>         | In this section you can configure the project source name. You can have more than one <code>[Client]</code> section. Additional client sections can be included as <code>[Client1]</code> , <code>[Client2]</code> , and so on.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>Client=</code>          | <p>Defines whether project sources are configured, as determined by the following values:</p> <ul style="list-style-type: none"> <li>1 : Configures project sources</li> <li>0 : Does not configure project sources</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>EncryptPassword=</code> | <p>Defines whether the password is encrypted in the response file, as determined by the following values:</p> <ul style="list-style-type: none"> <li>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.</li> <li>1 : Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</li> </ul> |

| Options         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DataSource=     | Defines the name of the new project source to create.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| ConnType=       | <p>Defines the database connection type for a project source. The following connection types are supported:</p> <ul style="list-style-type: none"> <li>2 : Connects a project source to the metadata using an ODBC DSN (Windows only).</li> <li>3 : Connects a project source to the metadata through a MicroStrategy Intelligence Server (three-tier).</li> </ul>                                                                                                                                                                                        |
| 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= | <p>The following authentication modes are supported:</p> <ul style="list-style-type: none"> <li>1 : Standard or login ID and password entered by the user</li> <li>2 : Network login ID (Windows authentication)</li> <li>8 : Guest account (Anonymous authentication)</li> <li>16 : LDAP authentication</li> <li>32 : Database login ID and password (database authentication)</li> <li>128 : Integrated authentication</li> </ul> <p>For information on the available authentication modes, see the <a href="#">Authentication modes, page 187</a>.</p> |
| MDPrefix=       | If using connection type 2 (ConnType=2) , defines a prefix for metadata repository tables.                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Timeout=        | <p>Defines and enforce a connection time out for inactive users connected to a project source. The following values are supported:</p> <ul style="list-style-type: none"> <li>0: Defines that users are not disconnected from project sources due to inactivity.</li> <li>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.</li> </ul>                                                                    |

# 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 203](#), [Creating a database connection, page 205](#), and [Creating a database login, page 210](#).

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](#).

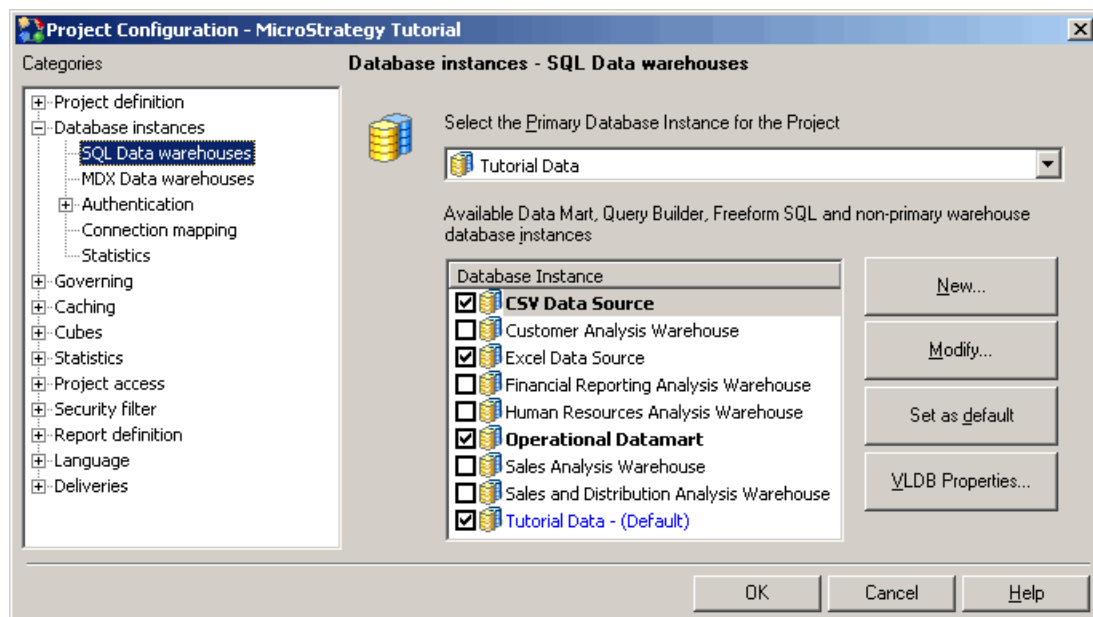
These non-project database instances can allow a project to connect to the data sources for the various features and additional data sources mentioned above, instead of accessing the data from the project's relational schema. For more information on the Warehouse Catalog, see the [Project Design Guide](#).

The database instances that you create are separated into two categories:


- [SQL data warehouses database instances, page 200](#)
- [MDX cube database instances, page 201](#)

## 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.

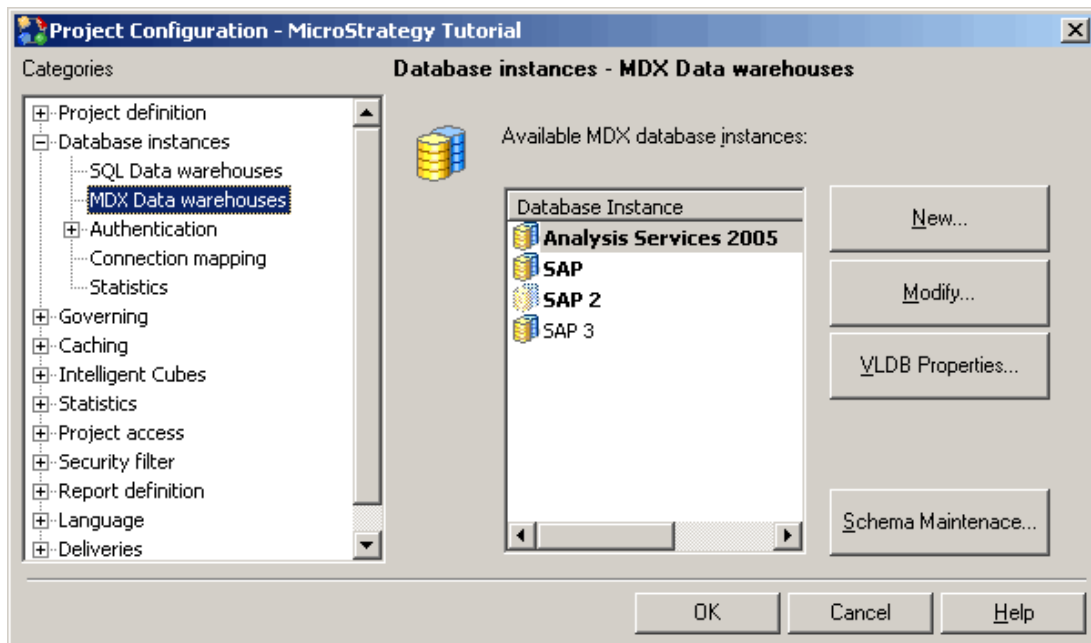
For information on the Warehouse Catalog as well as accessing multiple data sources with the MultiSource Option, see the [Project Design Guide](#).

- **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

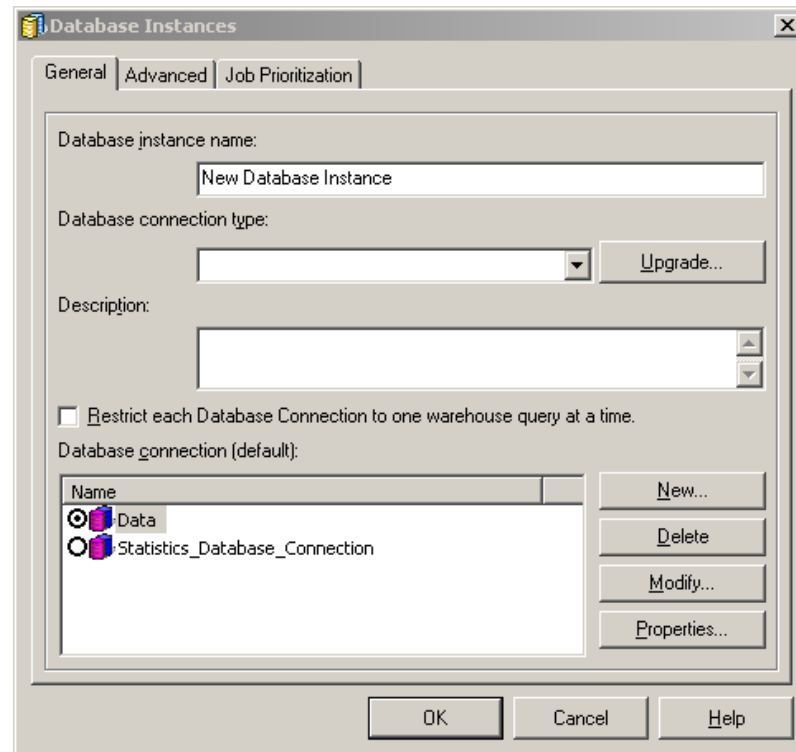
You can click Schema Maintenance to perform various tasks for an MDX cube schema that is part of your project, as described below:

- 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.



**i** 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.

**i** 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.

**i** 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 207](#).
  - **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**.

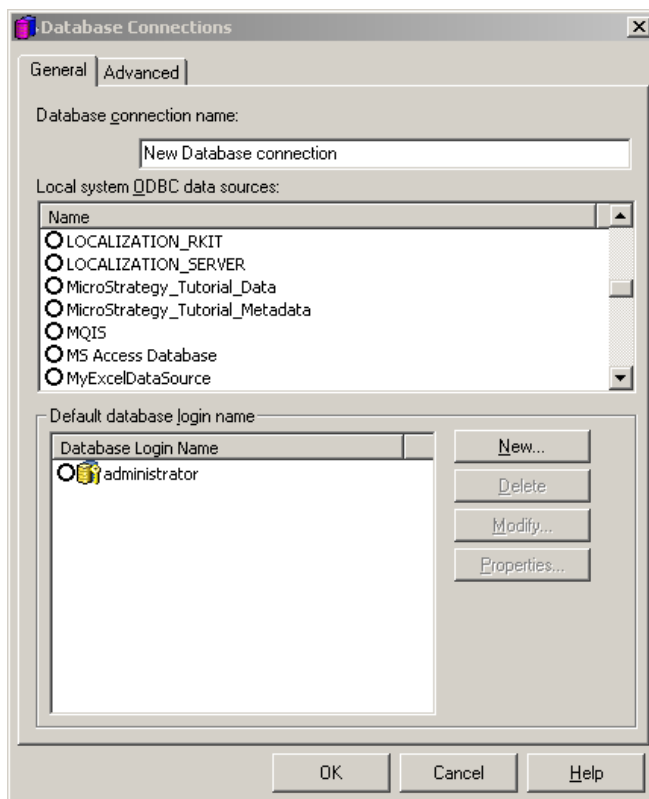
If the necessary database connection does not exist, you can create one by clicking **New**. For steps to create a database connection, see [Creating a database connection, page 205](#) below.

## 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 211](#).

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 Linux, page 408](#).

- **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 203.

---

## 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 204](#).



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 210](#) 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.

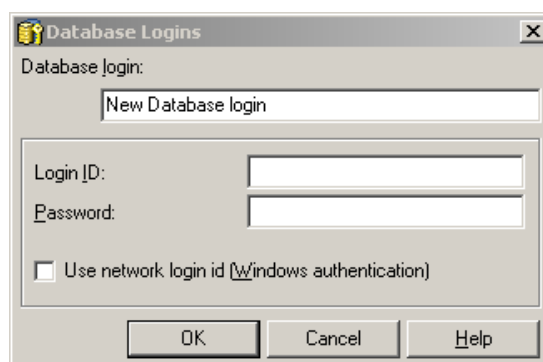
 Connection mapping is explained in [User connection mapping, page 211](#).

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 203](#).
- A database connection has been created, as described in [Creating a database connection, page 205](#).

## 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 203](#).
- A database connection has been created, as described in [Creating a database connection, page 205](#).

- A database login has been created, as described in [Creating a database login, page 210](#).

---

## 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 79](#) in [Chapter 1, Planning Your](#)

*Installation*, for an installation and configuration checklist.

# DEPLOYING MICROSTRATEGY WEB AND MOBILE SERVER

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

Steps to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP) in a Linux and Windows environment with various Web and application servers are also explained in detail. MicroStrategy Web (JSP) and Mobile Server (JSP) are platform-independent and can be deployed using different combinations of operating systems, Web servers, and application servers.

 Web application servers are not MicroStrategy products, so detailed steps cannot be provided for every combination of application server and operating system. This chapter supplies instructions for a few of the most common combinations. The procedures for different operating systems are similar, but you should refer to the vendor-provided information and documentation for details specific to your configuration, or contact MicroStrategy Technical Support.

MicroStrategy Web simplifies the job of deploying to large user groups because end users' machines only need a supported browser. MicroStrategy Web can be accessed from any supported browser because no code must be downloaded. Working as a thin client, MicroStrategy Web provides the functionality that end users and power users require to take full advantage of the MicroStrategy product suite.

This chapter has the following sections:

- [Deploying with IIS \(Windows\), page 215](#)



Deploying with IIS is the only setup given for the ASP.NET versions of MicroStrategy Web and Mobile Server. 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 218](#)
- [Deploying with WebLogic and Apache \(Solaris\), page 221](#)
- [Deploying with WebSphere and IBM HTTP Server \(AIX\), page 234](#)
- [Deploying with Oracle Glassfish Server \(Solaris\), page 243](#)

- [Deploying with Tomcat \(Windows\), page 250](#)
- [Deploying with Tomcat \(Linux\), page 255](#)
- [Deploying with SAP NetWeaver \(Windows\), page 259](#)
- [Deploying with Oracle 10g \(Windows\), page 262](#)
- [Deploying with JBoss \(Windows\), page 265](#)
- [Administering your MicroStrategy Web deployment, page 269](#)
- [Using absolute paths to share configuration files](#)
- [Configuring third-party data sources for importing data](#)
- [Configuring your MicroStrategy installation, page 273](#)

## Deploying with IIS (Windows)

Microsoft IIS can be used to deploy MicroStrategy Web and MicroStrategy Mobile Server:

### Deploying MicroStrategy Web

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

#### 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 64](#).
- 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
  - **64-bit Windows environments:** C:\Program Files (x86)\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

- 1 On the MicroStrategy Web Administrator page, click **Modify** in the Properties column of the Intelligence Server.
- 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 64](#).
- 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](#).

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

---

- 1 On the MicroStrategy Mobile Server Administrator page, click **Modify** in the Properties column of the Intelligence Server.
- 2 Select the **Automatically connect to Intelligence Server when Mobile Server or Intelligence Server is restarted** option.
- 3 Click **Save**.

# General steps to deploy MicroStrategy JSP applications

After you have installed MicroStrategy Web (JSP), Mobile Server (JSP) (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.

| Application                       | High-Level Deployment Steps                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MicroStrategy Web (JSP)           | <ol style="list-style-type: none"> <li>1 Log on to the application server by using the proper user name and password.</li> <li>2 Locate the <code>MicroStrategy.war</code> file in the MicroStrategy Web (JSP) Deployment Directory you specified during installation.</li> <li>3 To increase the performance of the application before proceeding with the deployment, see the <i>Performance-based setup information</i> 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.</li> <li>4 Choose the desired deployment method and follow the deployment procedure.</li> <li>5 Log on to the MicroStrategy Web Administrator Page.</li> <li>6 Launch MicroStrategy Web.</li> <li>7 Start working with the application.</li> </ol> <p> You must perform extra configuration steps to allow graphs to support non-Western European fonts on MicroStrategy Web (JSP) for a Linux system. For more information, see <a href="#">Graph and document support of non-Western European fonts</a>, page 442 of <a href="#">Appendix C, Troubleshooting</a>.</p> |
| MicroStrategy Mobile Server (JSP) | <ol style="list-style-type: none"> <li>1 Log on to the application server by using the proper user name and password.</li> <li>2 Locate the <code>MicroStrategyMobile.war</code> file in the MicroStrategy Mobile Server (JSP) Deployment Directory you specified during installation.</li> <li>3 To increase the performance of the application before proceeding with the deployment, see the <i>Performance-based setup information</i> 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.</li> <li>4 Choose the desired deployment method and follow the deployment procedure.</li> <li>5 Log on to the MicroStrategy Mobile Server Administrator Page.</li> <li>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 <a href="#">MicroStrategy Mobile Design and Administration Guide</a>.</li> <li>7 Start working with the application.</li> </ol>            |

## 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:

| Component and WAR File Name                                  | Default WAR File Location                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MicroStrategy Web (JSP)<br>MicroStrategy.war                 | <ul style="list-style-type: none"> <li>32-bit Windows environments:<br/>C:\Program Files\MicroStrategy\WebJSP</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\WebJSP</li> <li>Linux environments: <i>INSTALL_PATH</i>/WebUniversal</li> </ul> <p>Where <i>INSTALL_PATH</i> is the directory you specified as the MicroStrategy install directory during installation.</p>                 |
| MicroStrategy Mobile Server (JSP)<br>MicroStrategyMobile.war | <ul style="list-style-type: none"> <li>32-bit Windows environments:<br/>C:\Program Files\MicroStrategy\Mobile Server JSP</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Mobile Server JSP</li> <li>Linux environments: <i>INSTALL_PATH</i>/Mobile</li> </ul> <p>Where <i>INSTALL_PATH</i> is the directory you specified as the MicroStrategy install directory during installation.</p> |

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 219](#).

## 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

| Directory | Contents                              |
|-----------|---------------------------------------|
| \assets   | Supporting files                      |
| \html     | Supporting files                      |
| \images   | All image files                       |
| \import   | Sample files for Data Import analysis |

| Directory     | Contents                                        |
|---------------|-------------------------------------------------|
| \javascript   | Interface JavaScript files                      |
| \jsp          | Interface JSP code files                        |
| \META-INF     | Configuration files                             |
| \plugins      | Plug-in files for customizations                |
| \resBundles   | Flash descriptor files                          |
| \style        | Interface style files                           |
| \swf          | Supporting files for widgets                    |
| \VisFramework | Supporting files for visualizations             |
| \WEB-INF      | Configuration information for MicroStrategy Web |

## MicroStrategy Mobile Server

| Directory     | Contents                                                  |
|---------------|-----------------------------------------------------------|
| \assets       | Supporting files                                          |
| \css          | Supporting css files                                      |
| \html         | Supporting files                                          |
| \images       | All image files                                           |
| \import       | Sample files for Data Import analysis                     |
| \javascript   | Interface JavaScript files                                |
| \jsp          | Interface JSP code files                                  |
| \META-INF     | Configuration files                                       |
| \plugins      | Plug-in files for customizations                          |
| \style        | Interface style files                                     |
| \swf          | Supporting files for widgets                              |
| \ui           | Interface files                                           |
| \VisFramework | Supporting files for visualizations                       |
| \WEB-INF      | Configuration information for MicroStrategy Mobile Server |

## Deploying with WebLogic and Apache (Solaris)

This section provides information used to deploy and configure MicroStrategy JSP applications on the Oracle Solaris operating system, using Apache as the Web server and Oracle WebLogic Server as the application server. It provides information for WebLogic 10.3. You can also the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP).

This section includes the following information:

- [WebLogic paths and folder locations, page 221](#): Default folder structure for each version of WebLogic.
- [Preconfiguration information, page 222](#): Configuration that must occur before you begin deploying MicroStrategy Web (JSP) and Mobile Server (JSP).
- [Deploying MicroStrategy Web and Mobile Server, page 223](#): Instructions for deploying the application.
- [Re-deploy the application, page 230](#): Instructions for re-deploying the application.
- [Performance-based setup information, page 230](#): 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/`



- `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 218](#)
- [Setting up Apache Web server to proxy requests to WebLogic, page 222](#)

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.

| Requirement        | Recommended                                                                                                                                                    |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Operating system   | Oracle Solaris 10.x or 11.x (on SPARC)                                                                                                                         |
| Web server         | Apache 2.x                                                                                                                                                     |
| Application server | WebLogic 10.3                                                                                                                                                  |
| JDK                | Oracle JDK 1.6.0 or 1.7.0<br> You can download the JDK <a href="#">here</a> . |

- 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>
- 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>

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 230](#).

## Deploying MicroStrategy Web and Mobile Server

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

- 1 [Deploying automatically \(development mode\), page 223](#).  
- or -  
[Deploying manually \(production mode\), page 226](#).
- 2 [Configuring administrative access to MicroStrategy JSP applications, page 239](#).
- 3 [Launching the project, page 229](#).



The [Performance-based setup information, page 230](#) section provides information on additional settings to increase application performance. These additional settings are not required but can increase the performance of MicroStrategy Web (JSP). Review this information prior to deployment to see if these options are of interest to you.

You can deploy MicroStrategy Web and Mobile Server using one of the following deployment methods:

- The automatic deployment feature is the easiest and fastest way. See [Deploying automatically \(development mode\), page 223](#). 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 226](#).

### 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 218](#).
- 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 228](#).





To increase the performance of MicroStrategy Web JSP, you can configure additional settings after deployment. For more information, see [Performance-based setup information, page 230](#).

---

## To automatically deploy MicroStrategy JSP applications from an exploded directory

---



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

- 1 Locate the WAR file for your MicroStrategy JSP application, as described in [Locating the WAR file, page 218](#).

- 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 228](#).



To increase the performance of MicroStrategy Web (JSP), you can configure additional settings after deployment. For more information, see [Performance-based setup information, page 230](#).

## 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 218](#).

## 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:

```
#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

- 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 227.

---

### 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 218.
- 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 227 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 and Mobile Server as a WebLogic Server Administration Console web application, you have completed the steps required to deploy the application.

To launch the administrative page for MicroStrategy Web (JSP), Mobile Server (JSP) (JSP), see [Configuring administrative access to MicroStrategy JSP applications, page 228](#).



To increase the performance of MicroStrategy Web (JSP), you can configure additional settings after deployment. For more information, see [Performance-based setup information, page 230](#).

## Configuring administrative access to MicroStrategy JSP applications

Before you start MicroStrategy Web (JSP), Mobile Server (JSP) (JSP), you must configure their administrator pages.

### To configure access to the MicroStrategy JSP applications

- 1 The following table lists the URL to access MicroStrategy Web Administrator and MicroStrategy Mobile Server Administrator, for each deployment method.

The servlet names are case-sensitive. Make sure to use the correct case when typing the `mstrWebAdmin` name.

If the application server is enabled with security, a dialog box related to the administrator authentication opens.

| Deployment Method    | Address                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Automatic deployment | <p>Access the Administrator page from a web browser using this URL:</p> <ul style="list-style-type: none"> <li>• For Web (JSP): <code>http://IPAddress:7001/context_folder_Web/servlet/mstrWebAdmin</code></li> </ul> <p>In the URL listed above, <code>context_folder_Web</code> is the name of the folder where the Web (JSP) application was exploded and <code>IPAddress</code> is the IP address of your machine.</p> <ul style="list-style-type: none"> <li>• For Mobile Server (JSP): <code>http://IPAddress:7001/context_folder_Mobile/servlet/mstrWebAdmin</code></li> </ul> <p>In the URL listed above, <code>context_folder_Mobile</code> is the name of the folder where the Mobile Server application was exploded and <code>IPAddress</code> is the IP address of your machine.</p> |

| Deployment Method | Address                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Manual deployment | <p>Access the Administrator page from a browser using this address:</p> <ul style="list-style-type: none"> <li>For Web (JSP): <code>http://IPaddress:7001/Web_name/servlet/mstrWebAdmin</code></li> </ul> <p>In the URL listed above, <i>IPaddress</i> is the IP address of your machine. Replace the <i>Web_name</i> variable with the name you specified in the deployed name field when configuring Web (JSP) from WebLogic Server Administration Console, for example, MyWebApp.</p> <ul style="list-style-type: none"> <li>For Mobile Server (JSP): <code>http://IPaddress:7001/Mobile_name/servlet/mstrWebAdmin</code></li> </ul> <p>In the URL listed above, <i>IPaddress</i> is the IP address of your machine. Replace the <i>Mobile_name</i> variable with the name you specified in the deployed name field when configuring Mobile Server from WebLogic Server Administration Console, for example, MyMobileApp.</p> |

- 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).
- 4 For MicroStrategy Web (JSP) deployments, launch the MicroStrategy Web (JSP) project, as described in [Launching the project, page 229](#).

## 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).

| Deployment Method    | Address                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Automatic deployment | <p>Access MicroStrategy Web (JSP) from a web browser using this URL:</p> <p><code>http://IPaddress:7001/context_folder/servlet/mstrWeb</code></p> <p>where <i>context_folder</i> is the name of the folder where the application was exploded and <i>IPaddress</i> is the IP address of your machine.</p>                                                                              |
| Manual deployment    | <p>Access MicroStrategy Web (JSP) from a browser using the address:</p> <p><code>http://IPaddress/name/servlet/mstrWeb</code></p> <p>where <i>IPaddress</i> is the IP address of your machine. Replace the <i>name</i> variable with the name you specified in the deployed name field when configuring from WebLogic Server Administration Console, for example, <i>MyWebApp</i>.</p> |

## 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 230](#)
- [Precompiling JSP files, page 231](#)
- [Disable/relax auto-reload parameters, page 231](#)
- [Configuring Apache Web server to serve static files, page 233](#)

### 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 512MB, 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>
```

- 3 Save the file.

## Disable/relax auto-reload parameters

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

- [Set the `pageCheckSeconds` parameter, page 232](#)
- [Set the `WebLogic Reload Period` parameter, page 232](#)

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:

Value	Description
0	Pages are checked on every request.
-1	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.
n	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.

## 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:

```

<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:

```
<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 221](#) for information on default folder structure.

- 2 Repeat the previous step for the JavaScript and style sheet folders, replacing the word `images` in the previous code with the folder name where the JavaScript and style sheet files are located.
- 3 Change the `MatchExpression` parameter by typing `*/servlet/*` in the `MatchExpression` parameter. For example,

```
<IfModule mod_weblogic.c>
WebLogicHost 10.15.133.56
WebLogicPort 7001
MatchExpression */servlet/*
</IfModule>
```

- 4 Stop and start the Apache Web server using the commands `apachectl start` and `apachectl stop`.

The Web server now serves image (GIF), style sheet (CSS), JavaScript, and all other static files, thus reducing the load on the application server and increasing the application's performance.

## Deploying with WebSphere and IBM HTTP Server (AIX)

This section provides information used to deploy and configure MicroStrategy JSP applications on an AIX machine using the WebSphere Server and the IBM HTTP Web Server. You can use the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP).

This section includes the following information:

- [Locating the WAR file, page 218](#): This file is used as part of the steps for [Preparing for the application installation, page 236](#).
- [Deploying MicroStrategy Web and Mobile Server, page 234](#): Instructions for deploying the application.
- [Performance-based setup information, page 241](#): Optional settings to increase applications performance.

## Deploying MicroStrategy Web and Mobile Server

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

- 1 [Launching the WebSphere Administrative Console, page 235](#)
- 2 [Starting the WebSphere default application server, page 235](#)
- 3 [Installing the Enterprise Application, page 236](#)
- 4 [Regenerating plugin-cfg.xml, page 238](#)
- 5 [Restarting the application server, page 238](#)

**6** *Configuring administrative access to MicroStrategy JSP applications, page 239***7** *Launching the project, page 240*

The *Performance-based setup information, page 241* 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 218](#)) for the MicroStrategy JSP plaction 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 and MicroStrategy Mobile Server Administrator page 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 239](#).

---

### 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 236](#). 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 236](#). 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 236](#). For example, the default name of the WAR file, which is `MicroStrategyMobile`.



The servlet names are case-sensitive. Use the correct case when typing the `mstrWebAdmin` name. If the application server is enabled with security, a dialog box related to the administrator authentication opens.

- 2 Type the user ID and password assigned with the `admin` role.
- 3 After you are authenticated:
  - 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).
- 4 If you are deploying MicroStrategy Web (JSP), proceed to launch the project. For more information, see [Launching the project, page 240](#).

## Launching the project

After configuring the MicroStrategy Web Administrator page, you must follow the steps described in this section to launch your project.

### To launch the project

- 1 Start the Apache Web server by using the following command:

```
/usr/HTTPServer/bin/apachectl start
```

For example, if the IBM HTTP server is installed in the default location `/usr/IBMIHS`, then use the following command:

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

- 2 In a Web browser, specify the following URL:

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

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

```
http://IPAddress/ContextRoot/servlet/mstrWeb
```



In these addresses, *ContextRoot* is the name you provided for the context root for Web Module box on Preparing for the application installation page. For example, the default name of the WAR file, which is *MicroStrategy*. For more information, refer to [Preparing for the application installation, page 236](#).

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 241](#)
- [Precompiling JSP files, page 242](#)
- [Configuring the IBM HTTP Server to serve static files, page 242](#)

## 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:

```
./JspBatchCompiler.sh -enterpriseapp.name
ApplicationName -webmodule.name webModule
-cell.name cellName -node.name nodeName
-server.name serverName -keepgenerated
```

*TRUE*

- If the administrative server is running in a security enabled mode, you are prompted for the user ID and password to connect to the Admin server.

```
username userID
password password
```

For each JSP file compiled without error, the following message appears: `Code generation successful.`

## Configuring the IBM HTTP Server to serve static files

The IBM HTTP Server (Web server) is tuned to effectively serve static files. As a result, perceived performance is greatly enhanced if you configure the IBM HTTP Server to serve image, style sheet, and JavaScript files. This also reduces the load on the WebSphere Server so that it can handle only dynamic files while IBM HTTP Server handles static files. This requires that you do the following:

- Configure the application server level to serve the Java Server Pages (JSPs) and servlets, which are dynamic files and handled by WebSphere.
- Configure the Web server level to serve the images, JavaScripts, and style sheets, which are static files and handled by the IBM HTTP Server.

For more information, see the *IBM WebSphere Application Server* redbook covering System Management and Configuration. This book discusses separating static content from dynamic content.

## Deploying with Oracle Glassfish Server (Solaris)

This section provides information used to deploy and configure MicroStrategy JSP applications on an Oracle Glassfish Server 3.1.x in a Linux environment. You can use the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP).

- [Locating the WAR file, page 218](#).
- [Deploying MicroStrategy Web and Mobile Server, page 243](#): Instructions for deploying MicroStrategy Web (JSP), Mobile Server (JSP).
- [Undeploying MicroStrategy JSP applications, page 249](#): Instructions for undeploying MicroStrategy JSP applications.

### Deploying MicroStrategy Web and Mobile Server

Once your machine has been configured, you can deploy MicroStrategy Web (JSP, Mobile Server (JSP) (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 243](#)
- 2 [Deploying your MicroStrategy JSP application, page 244](#)
- 3 [Configuring administrative access to MicroStrategy JSP applications, page 246](#)
- 4 [Accessing the MicroStrategy JSP application administrative page, page 248](#)
- 5 [Connecting to the Web \(JSP\) project page, page 249](#)



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 218](#)) for your MicroStrategy JSP application to the same machine as the Oracle Glassfish Server, or to a location that is accessible to the Oracle Glassfish Server machine.

---

## To launch the Oracle Glassfish Server Administration Console

---

- 1 Navigate to the following directory in the command prompt:

```
InstallDir/bin
```

where *InstallDir* is the directory where you installed Oracle Glassfish Server.

- 2 Type the following command to start the domain:

```
asadmin start-domain --domaindir DomainDirectory
DomainName
```

where:

- *DomainDirectory* is the path you defined when creating the domain. You can remove the `--domaindir` option if the domain uses the default directory.
- *DomainName* is the name of the domain you created in the previous steps.

For example, to start domain1, which is the default domain, type the following command:

```
asadmin start-domain domain1
```

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

```
http://MachineName:PortNumber
```

where:

- *MachineName* is the IP address or the name of the machine where you installed Oracle Glassfish Server.
- *PortNumber* is the port number you provided when creating the domain. The default port number is 4848.

- 4 If prompted, type the user name and password that you provided when creating the domain.

## Deploying your MicroStrategy JSP application

After launching the Oracle Glassfish Server Administration Console, follow the steps below to deploy MicroStrategy JSP applications as a WAR file.

### Prerequisites

- Save the WAR file (see [Locating the WAR file, page 218](#)) to the same machine as the Oracle Glassfish Server, or to a location that is accessible to the Oracle Glassfish Server machine.

- Access to the administrative pages for MicroStrategy Web (JSP) and Mobile Server (JSP) can be granted by using the `admin` security role and the associated `mstradmin` group. Granting this access to users can be done within the Oracle Glassfish Server Administration Console after deploying the WAR file. While this default behavior supports most deployment requirements, if you have specific security requirements for your system, you must modify the security role details prior to deploying the WAR file, as described in [Deploying MicroStrategy Web and Mobile Server, page 243](#).

---

## 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`.

- 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. To do this, users need to be assigned to the `mstradmin` group, which is part of the `admin` security role.

Oracle Glassfish Server supports the following authentication realms out-of-the-box:

- File realm
- Administration realm
- Certificate realm



A realm, also called a security policy domain or security domain, is a scope over which a common security policy is defined and enforced by the security administrator of the security service. For more information, see the following resource:

[http://docs.oracle.com/cd/E18930\\_01/html/821-2435/ggkuk.html#gkbiy](http://docs.oracle.com/cd/E18930_01/html/821-2435/ggkuk.html#gkbiy)

In Oracle Glassfish Server, the file realm is the default realm. For controlling access to the Administration pages, you can create users and user groups and assign the `mstradmin` group to users in your security realm.

## To create users that are assigned to the mstradmin group in the file realm

- 1** In the Administration Console, from the Tree pane on the left, click **server (Admin Server)**. Ensure that the server is running or click **Start** to start the server.
- 2** From the Tree pane on the left, 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 and MicroStrategy Mobile Server Administrator.
  - **New Password:** The password used to authenticate a user.
  - **Confirm New Password:** A confirmation of the password, required when creating a new user.
- 6 Click **OK** 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 and Mobile Server Administrator. This provides administrative access without having to make any modifications to `glassfish-web.xml`. In these scenarios, you can use the steps provided in [Deploying your MicroStrategy JSP application, page 244](#) and [Configuring administrative access to MicroStrategy JSP applications, page 246](#) 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 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 244](#)) and assign the security role to the necessary user accounts (see [Configuring administrative access to MicroStrategy JSP applications, page 246](#)).

## 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 244](#). For example, you can use the default name of the WAR file, which is *MicroStrategy*. The default port number is 8080.

- For Mobile Server (JSP):  
`http://  
IPAddress  
:PortNumber/ContextRootMobile/servlet/mstrWebAdmin`

In the URL listed above, *ContextRootMobile* is the name you provided for the *ContextRoot* for Web Module box in the section [Deploying your MicroStrategy JSP application, page 244](#). For example, you can use the default name of the WAR file, which is *MicroStrategyMobile*. The default port number is 8080.



The servlet names are case-sensitive. Use the correct case when typing the *mstrWebAdmin* name. If the application server is enabled with security, a dialog box related to the administrator authentication opens.

- 2** Type the user ID and password for a user who is a member of the *mstradmin* group, as described in [Configuring administrative access to MicroStrategy JSP applications, page 246](#).
- 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).
- 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 249](#) 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 below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

- [Preconfiguration information, page 250](#): Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP).
- [Deploying MicroStrategy Web and Mobile Server, page 252](#): 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.

### 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> 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:



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** field, type **JAVA\_HOME**.
- 4 In the **Variable Value** field, type the path of the folder where you installed the JDK and click **OK**.

For example, if the fully qualified path to your JDK executable is  
C:\jdk1.6.0\bin\java.exe, the value of your JAVA\_HOME variable is  
C:\jdk1.6.0.



If you have installed JDK under the `Program Files` folder, type `Program~1` when specifying the folder name in the Variable Value box; otherwise the system does not recognize the folder. For example, type `C:\Program~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:



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** 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 and Mobile Server

Assuming you have made all the necessary configurations described above, you can begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) with Tomcat. This involves the following steps:

- 1 *Deploying using Tomcat as a stand-alone Web container, page 252*
- 2 *Configuring administrative access your MicroStrategy JSP applications, page 253*
- 3 *Accessing the MicroStrategy JSP application administrative page, page 254*
- 4 *Launching the project, page 255*

## 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 218*.
- 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.

- 5 Browse to the `Tomcat\bin` folder, where `Tomcat` is the folder in which you installed Tomcat. For example, in the command prompt, type

```
cd C:\Tomcat\bin
```

- 6 Press ENTER.

`C:\Tomcat\bin>` is displayed at the command prompt.

- 7 Type the required commands to start and stop Tomcat, which depends on your version of Tomcat. For example, for Tomcat 7, type `Tomcat7 start` to start Tomcat and type `Tomcat7 stop` to stop Tomcat. Refer to your third-party Apache documentation for information on the commands to start and stop Tomcat.



If you installed Tomcat under the `Program Files` folder, type **Progra~1** when you change folders in the command prompt. Otherwise, the system does not recognize the folder. For example, type `C:\Progra~1\Tomcat\bin` in the command prompt.

Your MicroStrategy JSP application is deployed automatically, based on the following:

- If you have configured Tomcat to deploy an exploded WAR file, which is often the default behavior, a folder is created within the `Tomcat\webapps` folder:
  - When deploying MicroStrategy Web (JSP), the folder is named `MicroStrategy` by default.
  - When deploying MicroStrategy Mobile Server (JSP), the folder is named `MicroStrategyMobile` by default.
- If you have configured Tomcat to deploy an unexploded WAR file, the configuration files are created within the system's default temporary file directory. For Windows systems, the temporary file directory is commonly defined by the `TMP` environment variable:
  - When deploying MicroStrategy Web (JSP), a `/microstrategy/web-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Web (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.
  - When deploying MicroStrategy Mobile Server (JSP), a `/microstrategy/mobile-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Mobile Server (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.

## Configuring administrative access your MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must create the users and assign them the role of `admin` under the Tomcat user configuration file. The steps to configure this access are below.

## To configure administrative access to your MicroStrategy JSP applications

- 1 In the `Tomcat\conf` folder, open the `tomcat-users.xml` file in a program that allows you to edit the file, such as Notepad.
- 2 Add the following tag and save the file:

```
<user name="administrator" password="administrator"
roles="admin"/>
```



You can specify any value in the `user name` and `password` fields.

- 3 Stop and start Tomcat from the command line.

Now you can access and configure your MicroStrategy JSP application, as described in [Accessing the MicroStrategy JSP application administrative page, page 254](#).

## 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`

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).
- 4 If you are deploying MicroStrategy Web (JSP), proceed to launch the MicroStrategy project. For more information, see [Launching the project, page 255](#).

## Launching the project

In a Web browser, access MicroStrategy Web (JSP) using this URL:

`http://localhost:8080/MicroStrategy/servlet/mstrWeb`

## Deploying with Tomcat (Linux)

This section provides information on how to deploy and configure MicroStrategy JSP applications with Tomcat in a Linux environment. You can use the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

- [Preconfiguration information, page 255](#): Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP).
- [Deploying MicroStrategy Web and Mobile Server, page 257](#): 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 255](#)
- [Configuring the JDK, page 256](#)
- [Installing Tomcat, page 256](#)
- [Configuring Tomcat, page 256](#)
- [Locating the WAR file, page 218](#)

## 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> 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
```

## Configuring 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>
- Tomcat 8.0.x: website. <http://tomcat.apache.org/download-80.cgi>



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 `JAVA_OPTS` parameter in the `catalina.sh` file. For example, you can define this parameter as follows:

```
JAVA_OPTS = "-Xms1024m -Xmx2048m"
```

This value may need to be modified to reflect the requirements of your specific environment. Refer to your third-party application server documentation for information on how to determine a satisfactory Java heap size for your environment.

## Deploying MicroStrategy Web and Mobile Server

After you have performed the configurations described above, you can begin deploying MicroStrategy JSP applications with Tomcat. This involves the following steps:

- 1 [Deploying using Tomcat as a standalone Web container, page 257](#)
- 2 [Accessing the MicroStrategy JSP application administrative page, page 259](#)

## 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 218](#).
- 2 Copy the WAR file to the *Tomcat/webapps* directory.

## To start and stop Tomcat from the command line

- 3 Type # `$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 `Tomcat/webapps` folder:
    - When deploying MicroStrategy Web (JSP), the folder is named `MicroStrategy` by default.
    - When deploying MicroStrategy Mobile Server (JSP), the folder is named `MicroStrategyMobile` by default.
  - If you have configured Tomcat to deploy an unexploded WAR file, the configuration files are created within the system's default temporary file directory. For Linux systems, the temporary file directory is usually `/tmp/` or `/var/tmp/`:
    - When deploying MicroStrategy Web (JSP), a `/microstrategy/web-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Web (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.
    - When deploying MicroStrategy Mobile Server (JSP), a `/microstrategy/mobile-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Mobile Server (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.

## Configuring administrative access to MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must create the users and assign them the role of `admin` under the Tomcat user configuration file. The steps to configure this access are below.

## 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:
 

```
<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 259](#).

## 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`

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).

**4** If you are deploying MicroStrategy Web (JSP), proceed to launch the MicroStrategy project. In a Web browser, access MicroStrategy Web (JSP) using the following URL:

`http://localhost:8080/MicroStrategy/servlet/mstrWeb`

## Deploying with SAP NetWeaver (Windows)

This section provides information used to deploy and configure MicroStrategy JSP applications on a Windows machine using the SAP application server. You can use the procedure below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

*Deploying MicroStrategy Web and Mobile Server, page 260*: Instructions for deploying the application.

## Deploying MicroStrategy Web and Mobile Server

Once your machine has the necessary settings configured, you can deploy MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) on the SAP-Windows machine. Deployment involves the following steps:

- 1 *Deploying MicroStrategy JSP applications with the SAP NetWeaver Application Server, page 260*
- 2 *Configuring administrative access to MicroStrategy JSP applications, page 261*
- 3 *Accessing the MicroStrategy JSP applications, page 261*

## 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 218*.
- 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, 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 261](#).

---

## To access the MicroStrategy Web Administrator or Mobile Server Administrator page

---

- 1 Access the servlet by typing the following URL in a Web browser:
  - For Web (JSP):  
`http://  
MachineName:PortNumber/MicroStrategy/servlet/mstrWebAdmin`
  - For Mobile Server (JSP):  
`http://  
MachineName  
:PortNumber/MicroStrategyMobile/servlet/mstrWebAdmin`



The servlet names at the end of the URLs listed above are case-sensitive. Use the correct case when typing the servlet name.

The login dialog box opens.

- 2 Specify a user name and password.
- 3 After you are authenticated:
  - If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.
  - If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click **Mobile Configuration** to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the [MicroStrategy Mobile Design and Administration Guide](#). Creating a configuration completes the steps required to deploy Mobile Server (JSP).
- 4 If you are deploying MicroStrategy Web (JSP), access the MicroStrategy Web Application on SAP Web Server by specifying the following URL in the Web browser:

```
http://
MachineName:PortNumber/MicroStrategy/servlet/mstrWeb
```

## Deploying with Oracle 10g (Windows)

This chapter provides information used to deploy and configure MicroStrategy JSP applications with Apache as the Web server and Oracle Application Server 10g R3 as the application server. You can use the procedure below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

[Deploying MicroStrategy Web and Mobile Server, page 262](#): Instructions for deploying, including detailed steps.

## Deploying MicroStrategy Web and Mobile Server

After your machine is configured, you can start the deployment of your MicroStrategy JSP application with Oracle Application Server 10g R3.

To deploy MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) perform the following procedures:

- 1 [Deploying using the Oracle Enterprise Manager, page 262](#)
- 2 [Accessing the MicroStrategy JSP application administrative page, page 259](#)

## 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 218](#).
- 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.
- 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, 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 264](#).

## Accessing the MicroStrategy JSP administrative pages

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

---

### To access the MicroStrategy JSP administrative pages

---

- 1 In a Web browser, access the administrative page by specifying the following URL:

- For Web (JSP):  
`http://  
 IPAddress:PortNumber/MicroStrategy/servlet/mstrWebAdmin`
- For Mobile Server (JSP):  
`http://  
 IPAddress:PortNumber  
 /MicroStrategyMobile/servlet/mstrWebAdmin`

Where *IPAddress* is the IP address of the Oracle machine and *PortNumber* is the port number used by the Oracle Application Server. The servlet name at the end of the URLs listed above are case-sensitive, so be sure to use the correct case when typing the servlet name.

- 2 When prompted for a user name and password, specify the values you used earlier when creating the user mapped to the admin security role (see [Deploying using the Oracle Enterprise Manager, page 262](#) above).
- 3 After you are authenticated:
  - If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.
  - If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click **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).

- 4 If you are deploying MicroStrategy Web (JSP), you can now launch the MicroStrategy project. In a Web browser, access MicroStrategy Web (JSP) using this URL:

```
http://
IPAddress:PortNumber/MicroStrategy/servlet/mstrWeb
```

Where *IPAddress* is the IP address of the Oracle machine and *PortNumber* is the port number used by the Oracle Application Server.

## Deploying with JBoss (Windows)

This chapter provides information used to deploy and configure MicroStrategy JSP applications in a JBoss environment. You can use the steps below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

- [Preconfiguration information, page 265](#): configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP).
- [Deploying MicroStrategy Web and Mobile Server, page 266](#): instructions for deploying, including step-by-step procedures.

### Preconfiguration information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on JBoss on your machine.

#### Configuring the JDK

If you have not installed Oracle JDK yet, download them from the <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:\j sdk1.8.0\bin\java.exe`, the value of your `JAVA_HOME` variable is  
`C:\j sdk1.8.0`.



If you have installed JDK under the `Program Files` folder, type `Progra~1` in the destination folder; otherwise the system does not recognize the folder. For example,  
`C:\Progra~1\j sdk1.8.0`.

## Installing JBoss

You can download and install JBoss from the following website:

<http://www.jboss.org/jbossas/downloads>

Keep track of the location in which you install JBoss, as this location is used later (referred to as `JBOSS_HOME`) to configure JBoss with a MicroStrategy JSP application deployment.

## Deploying MicroStrategy Web and Mobile Server

Assuming you have made all the necessary configurations described above, you can begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) with JBoss. This involves the following steps:

- 1 *Deploying using JBoss as a stand-alone Web container, page 267*
- 2 *Configuring administrative access to MicroStrategy JSP applications, page 267*
- 3 *Accessing the MicroStrategy JSP application administrative page, page 268*

## 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 218](#).
- 2 Copy the WAR file to the `JBOSS_HOME\server\default\deploy` directory.
- 3 To start JBoss, browse to `JBOSS_HOME\bin`. Then run the following command:  

```
run.bat -b 0.0.0.0
```

Your MicroStrategy JSP application is deployed automatically, based on the following:

- If you have configured JBoss to deploy an exploded WAR file, which is often the default behavior, a folder is created within the `JBOSS_HOME\server\default\deploy` directory:
  - When deploying MicroStrategy Web (JSP), the folder is named `MicroStrategy` by default.
  - When deploying MicroStrategy Mobile Server (JSP), the folder is named `MicroStrategyMobile` by default.
- If you have configured JBoss to deploy an unexploded WAR file, the configuration files are created within the system's default temporary file directory. For Windows systems, the temporary file directory is commonly defined by the `TMP` environment variable:
  - When deploying MicroStrategy Web (JSP), a `/microstrategy/web-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Web (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.
  - When deploying MicroStrategy Mobile Server (JSP), a `/microstrategy/mobile-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Mobile Server (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.

## Configuring administrative access to MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must create the users and assign them the role of `admin` under the JBoss user configuration files. The steps to configure this access are below.

## To configure administrative access to MicroStrategy JSP applications

- 1 Browse to the directory `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 and Mobile Server Administrator, using the following syntax:

```
user_id=user_password
```

For example, you create UserA and UserB with passwords 1234 and 5678 respectively using the following syntax:

```
UserA=1234
```

```
UserB=5678
```

- 5 Save your changes and close the `users.properties` file.

- 6 Open the `roles.properties` file in a text editor.

- 7 Include one line for each user you included in the `users.properties` file and grant them administrative access, using the following syntax:

```
user_id=admin
```

For example, you define UserA and UserB to have administrative access using the following syntax:

```
UserA=admin
```

```
UserB=admin
```

- 8 Save your changes and close the `roles.properties` file.

- 9 To start JBoss, browse to `JBOSS_HOME\bin`. Then run the following command:

```
run.bat -b 0.0.0.0
```

Now you can access and configure your MicroStrategy JSP application, as described in [Accessing the MicroStrategy JSP application administrative page, page 268](#).

## 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`

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).

**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

- You must install and deploy MicroStrategy Web Services to allow the installation of MicroStrategy Office from MicroStrategy Web. For information about deploying MicroStrategy Web Services, see the [MicroStrategy Office User Guide](#).

## 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**.

- 2 Underneath Web Server on the left, click **MicroStrategy Office**. The MicroStrategy Office settings page opens.

- 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 `/Lang_xxxx/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 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:

```
<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:

```
<parameter name="serverConfigFilesDefaultLocation"
value="ABSOLUTE:/usr/User1/MicroStrategy/xml/" />
<parameter name="serverLogFilesDefaultLocation"
value="ABSOLUTE:/usr/User1/MicroStrategy/xml/log/" />
```

## Configuring third-party data sources for importing data

You can use MicroStrategy Web to import data from different data sources, such as an Excel file, a table in a database, the results of a Freeform SQL query, or other data sources, into MicroStrategy metadata, with minimum project design requirements.

To import data from the following data sources, configure a secure connection between your third-party data source and MicroStrategy Web:

- Dropbox
- Google Analytics
- Google BigQuery
- Google Drive
- Facebook
- Salesforce.com
- Twitter

The steps below show you how to make these third-party data sources available for import into MicroStrategy Web.

## 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 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 407](#).

## To configure a connection between a third-party data source and MicroStrategy Web for Data Import

- 1 Access the administrative options for your third-party data source. For example, if you are integrating data from Salesforce.com, log in to Salesforce.com by accessing <https://login.salesforce.com/>.
- 2 You must configure MicroStrategy Web as a remote access application for the third-party data source. For the steps to define remote access applications in your third-party data source, refer to your third-party documentation.

When configuring MicroStrategy Web as a remote access application, you must define the callback URL as the URL to access MicroStrategy Web, including the event 3172. Depending on how you deployed MicroStrategy Web, the syntax for this URL can take one of the following forms:

- For MicroStrategy Web (ASP.NET) deployments:

```
https://
WebServer
/
WebApplicationName
/asp/Main.aspx?evt=3172&src=Main.aspx.3172
```

- 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 and Mobile Server, you can continue your setup and configuration. To help guide the rest of your installation and configuration steps, refer to the section [Installation and configuration checklists, page 79](#) in [Chapter 1, Planning Your Installation](#), for installation and configuration checklists.

# SETTING UP DOCUMENTS AND HTML DOCUMENTS

This chapter explains the setup required for Intelligence Server to execute HTML documents and Report Services documents on Linux platforms.

This chapter includes the following sections:

<a href="#">Prerequisites</a>	274
<a href="#">Executing documents and HTML documents in Linux</a>	275
<a href="#">Configuring your MicroStrategy installation</a>	280

## Prerequisites

This chapter assumes the following:

- You are familiar with MicroStrategy Developer and MicroStrategy Intelligence Server.
- You are familiar with MicroStrategy HTML documents and Report Services documents.
- You have a Report Services product license if you are using Report Services documents. HTML documents do not require a Report Services product license.
- You have installed MicroStrategy Developer on a Windows machine.
- You have installed MicroStrategy Intelligence Server on a Linux machine.
  - Some of the steps described in this document may require root access permissions.
-  You must perform extra configuration steps to allow Report Services documents to support non-Western European fonts on a UNIX system. For more information, see [Graph and document support of non-Western European fonts](#), page 442 of [Appendix C, Troubleshooting](#).

## Executing documents and HTML documents in Linux

A MicroStrategy Report Services document contains objects representing data coming from one or more reports, as well as positioning and formatting information. Report Services documents help format data from multiple reports in a single display and can be used for presentations. When you create a document, you can specify the data that appears, control the layout, formatting, grouping, and subtotaling of data and specify the position of page breaks. In addition, you can insert pictures and draw borders in the document.



In this 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 Linux platform, but you must share the directory with the Windows platform that includes MicroStrategy Developer. For more information on setting up the HTML document directory on a Linux platform, see [Setup for creating and executing HTML documents, page 275](#).



For additional information on HTML documents, see the *HTML Documents* chapter in the [Advanced Reporting Guide](#).

## Setup for creating and executing HTML documents

HTML documents can only be created with MicroStrategy Developer on a Windows platform, but they can be stored and executed from a directory within a Linux platform. The directory that stores the HTML documents must be accessible on the computer with Intelligence Server and the Windows computer with Developer.

Using the Project Configuration dialog box in MicroStrategy Developer, you must specify the location of the HTML document directory as an absolute path. This document directory can be on a local machine or on a remote machine. Users require appropriate read and write permissions to access this directory. When MicroStrategy Intelligence Server executes HTML documents, it requires read permission to the HTML document directory to access the HTML files.

For the procedure of setting up an HTML document directory between Windows and Linux computers below, the following assumptions are made:

- You have installed MicroStrategy Developer on a Windows computer and installed MicroStrategy Intelligence Server on a Linux computer.



Developer can only be installed on a Windows computer.

- MicroStrategy Developer users have at least read permissions to the HTML document directory for executing existing HTML documents. Write permissions to the directory are required for MicroStrategy Developer users to create new HTML documents.
- For the file paths described in the procedure below, *machine-name* is used to represent the name of the machine you store the HTML document directory on. For example, if you store the directory on a machine named UNIX1, *machine-name* should be replaced with UNIX1. This machine must have Samba installed to provide access to the folder on a Windows computer.
- You must have root permissions on any Linux computer used to set up the HTML document directory. This includes the computer that stores the HTML document directory as well as any computer that must be setup to access the directory.

---

## To set up the HTML document directory

---

- 1 Create a directory to hold the HTML document directory on the desired Linux computer. This procedure assumes that the path of the HTML document directory is *machine-name*: /share/htmldocuments. This is the machine that is referenced as *machine-name* in the steps below. To create this directory, enter the commands below:

```
cd /

mkdir share

cd share

mkdir htmldocuments
```

- 2 Install Samba software on the Linux computer that you created the HTML documents directory in the step above. With this software, the HTML documents directory is accessible to the Windows computer with MicroStrategy Developer installed. You can get the latest version of Samba at <http://www.samba.org>.



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 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>.

- 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`.

```
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 Linux HTML documents directory from the Windows computer so that you are not prompted for authentication each time the directory is accessed:
  - a From the Windows computer that has MicroStrategy Developer installed, click **Start**, and select **Run**. The Run dialog box opens.
  - b Type `\\machine-name\share\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.

- 8 Using the Project Configuration dialog box in MicroStrategy Developer, set the HTML document directory as an absolute path by following the steps below:
  - a In Developer, right-click the project associated with the HTML documents and select **Project Configuration**. The Project Configuration dialog box opens.
  - b Expand **Project definition** and click **Advanced**. The Project Configuration - Advanced options are displayed.
  - c In the HTML document directory box, type the absolute path `\\machine-name\share\htmldocuments`. For example, type `\\UNIX1\share\htmldocuments`.
  - d Click **OK** to accept the changes.

- 9 Create a directory named **xsls** under the HTML document directory and copy the XSL files you require for creating HTML documents to the **xsls** directory, */machine-name/share/htmldocuments/xsls*. If you stored XSL files in a different directory or did not copy them from their original default directory, you must copy them into the new **xsls** directory. For example, the default HTML document directory for the Tutorial project is `Program Files\MicroStrategy\Tutorial Reporting`.
- 10 If you want to insert images into the HTML document, create a directory named **images** under the HTML document directory, and copy the images to the directory */machine-name/share/htmldocuments/images*.

You are now ready to create and execute your HTML documents. Remember to create your HTML documents in the HTML document directory, otherwise, Intelligence Server cannot execute the HTML documents correctly.

## Setup for executing existing HTML documents

If you have created HTML documents prior to establishing a connection between the HTML document directory on the Linux machine with MicroStrategy Intelligence Server and the Windows machine with MicroStrategy Developer, you must make sure that all the files used for an HTML document are copied to the shared HTML document directory. After the connection is established, you should always create the HTML documents in the shared HTML document directory. Once the existing files are copied, you can execute the HTML documents using Intelligence Server.

In the procedure of setting up existing HTML documents, the following assumptions are made:

- You have completed all the steps listed in the section [Setup for creating and executing HTML documents, page 275](#).
- 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 Linux environment, you must perform some additional setup tasks.

When Intelligence Server is running on a Linux platform, all fonts are converted to the Courier New font for:

- Reports exported to PDF format
- Report Services documents
- Graphs contained in HTML documents
- Graphs displayed in MicroStrategy Web

This occurs because the fonts required by the PDF component are missing from Linux machines running Intelligence Server. The missing fonts may include Microsoft True Type fonts.



MicroStrategy does not distribute or license Microsoft fonts, and therefore cannot package Microsoft fonts with Intelligence Server.

To resolve this issue, you must install the font files in the `PDFGeneratorFiles` folder within the MicroStrategy installation path on the Linux machine, as described below.

---

### To copy fonts to your Linux machine

---

- 1 Log in to your Linux machine that hosts Intelligence Server.
- 2 Install the Microsoft True Type fonts. Refer to the following resources for information on licensing requirements for and installing Microsoft True Type fonts:
  - <http://www.microsoft.com/typography/RedistributionFAQ.msp>
  - [http://www.ascendercorp.com/msfonts/msfonts\\_main.html](http://www.ascendercorp.com/msfonts/msfonts_main.html)
  - <http://corefonts.sourceforge.net/>

- 3 Copy the font files into the `INSTALL_PATH/PDFGeneratorFiles` directory, where `INSTALL_PATH` is the directory you specified as the MicroStrategy install directory during installation.
- 4 To update the list of fonts available, you must restart the Intelligence Server.

## Configuring your MicroStrategy installation

After completing the steps to set up documents and HTML documents for Linux, you can continue your setup and configuration. To help guide the rest of your installation and configuration steps, refer to the section [Installation and configuration checklists, page 79](#) 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"`
- Response file in **[LogFile]**. See [Configuring a response.ini file to install MicroStrategy, page 283](#) for more information.

## Methods of installation

The installation methods discussed in this chapter are:

- [Installing and configuring with a response.ini file, page 283](#)
- [Silent installation, page 324](#)

# 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 283](#)
- [Configuring your installation with a response.ini file, page 322](#)
- [Uninstalling with a response.ini file, page 322](#)

## 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 324](#).

## Component dependencies

When you use a `response.ini` file to install MicroStrategy products, there are some key dependencies among separate components you should be aware of. The products listed below require either pre-installed software or certain MicroStrategy components to be selected to successfully install the products with a `response.ini` file:

- MicroStrategy Enterprise Manager requires MicroStrategy Command Manager to be included in the installation.
- MicroStrategy Delivery Engine, MicroStrategy Subscription Portal, MicroStrategy Tutorial – Delivery Installation and MicroStrategy Tutorial – Delivery Configuration 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 Function Plug-in Wizard requires Microsoft Visual C++ to be installed before running your `response.ini` file.
- MicroStrategy Analytic Modules requires MicroStrategy Developer or a combination of MicroStrategy Analyst and MicroStrategy Architect 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 327](#).
- 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.



The options are case sensitive, therefore they must be entered as indicated in the tables below.

### Installer

Options	Description
[Installer]	Section that begins the installation.

Options	Description
ExpressMode =	TRUE or False. Indicates whether the installation uses Express Install or Custom Install.
HideAllDialogs =	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.
LogFile =	Location where the <code>install.log</code> file is generated. If left empty, it takes the default location and file name of: <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\Common Files\MicroStrategy\install.log.</li> <li>64-bit Windows environments: C:\Program Files (x86)\Common Files\MicroStrategy\install.log.</li> </ul>
CreateShortcuts =	TRUE or FALSE. TRUE creates the shortcuts for MicroStrategy products, tools, and documentation. The default is TRUE.
PreventReboot =	TRUE or FALSE. TRUE prevents the machine from rebooting after installation is done. Note the following conditions: <ul style="list-style-type: none"> <li>If both ForceReboot = TRUE and PreventReboot = TRUE, then PreventReboot applies first.</li> <li>If both ForceReboot and PreventReboot are FALSE and HideDialog for [Finish] is set to TRUE, then the machine reboots only if it is required.</li> </ul> The default is FALSE.
ForceReboot =	TRUE or FALSE. TRUE reboots the machine after the installation is done. The default is FALSE.
EnterpriseManagerOverwrite =	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.
RunConfigWizard =	TRUE or FALSE. When using silent install, set to FALSE to prevent the Configuration Wizard from coming up after reboot. The default is TRUE.
ConfigWizardResponseFile =	Specify the name of the response file for the Configuration Wizard; otherwise, it takes the default name of <code>response.ini</code> .  For more details on configuring the <code>response.ini</code> file for the Configuration Wizard, see <a href="#">Configuring your installation with a response.ini file, page 322</a> .
BackupFiles =	TRUE or FALSE. If you set the value to TRUE, it creates a backup of

Options	Description
	<p>the following files:</p> <ul style="list-style-type: none"> <li>• *.pds</li> <li>• *.xsl</li> <li>• *.asp</li> <li>• *.css</li> <li>• *.js</li> <li>• *.sql</li> </ul> <p>The default is FALSE.</p>
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 =	<p>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:</p> <ul style="list-style-type: none"> <li>• Reboot at the beginning of installation to replace this file. It is recommended that you select this option.</li> <li>• Continue with the installation at the risk of the software not functioning properly.</li> </ul> <p>If you enter FALSE, the prompt does not display. The default is TRUE.</p>
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.
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.

Options	Description
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.
COMMONFILES =	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: <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\Common Files\MicroStrategy</li> <li>64-bit Windows environments: C:\Program Files (x86)\Common Files\MicroStrategy</li> </ul>
Developer =	Location where Developer will be installed. If left empty, it takes the default location of: <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Desktop</li> <li>64-bit Windows environments: C:\Program Files</li> </ul>

Options	Description
	(x86) \MicroStrategy\Desktop
ObjectManager =	<p>Location where Object Manager will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Object Manager</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Object Manager</li> </ul>
CommandManager =	<p>Location where Command Manager will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Command Manager</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Command Manager</li> </ul>
EnterpriseManager =	<p>Location where Enterprise Manager will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Enterprise Manager</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Enterprise Manager</li> </ul>
Server =	<p>Location where the MicroStrategy Intelligence Server will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Intelligence Server</li> </ul>
Web =	<p>Location where MicroStrategy Web will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Web ASPx</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Web ASPx</li> </ul>
WebUniversal =	<p>Location where MicroStrategy Web will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Web JSP</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Web JSP</li> </ul>



Options	Description
WebServices =	<p>Location where MicroStrategy Web Services will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Web Services</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Web Services</li> </ul>
WebServicesUniversal =	<p>Location where MicroStrategy Web Services will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Web Services JSP</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Web Services JSP</li> </ul>
Office =	<p>Location where MicroStrategy Office will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Office</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Office</li> </ul>
TutorialReporting =	<p>Location where MicroStrategy Tutorial - Reporting will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Tutorial Reporting</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Tutorial Reporting</li> </ul>
AnalyticsModules =	<p>Location where the Analytics Modules will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Analytics Modules</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Analytics Modules</li> </ul>
NCSAdminDeliveryEngine =	<p>Location where the Narrowcast Server Delivery Engine will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Narrowcast Server\Delivery Engine</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Narrowcast Server\Delivery Engine</li> </ul>

Options	Description
SubscriptionPortal =	<p>Location where the Subscription Portal will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Narrowcast Server\Subscription Portal</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Narrowcast Server\Subscription Portal</li> </ul>
TutorialDelivery =	<p>Location where MicroStrategy Tutorial - Delivery will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Narrowcast Server\Tutorial</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Narrowcast Server\Tutorial</li> </ul>
IntegrityManager =	<p>Location where MicroStrategy Integrity Manager will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Integrity Manager</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Integrity Manager</li> </ul>
Mobile =	<p>Location where MicroStrategy Mobile will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Mobile Clients</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Mobile Clients</li> </ul>
MobileASPPath =	<p>Location where MicroStrategy Mobile Server ASP.NET will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Mobile Server ASPx</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Mobile Server ASPx</li> </ul>
MobileJSPPath =	<p>Location where MicroStrategy Mobile Server JSP will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Mobile Server JSP</li> <li>64-bit Windows environments: C:\Program Files</li> </ul>

Options	Description
	(x86)\MicroStrategy\Mobile Server JSP
Portlets =	<p>Location where MicroStrategy Portlets will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\Portlets</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\Portlets</li> </ul>
MDXCubeProvider =	<p>Location where the MicroStrategy MDX Cube Provider will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\MDX Cube Provider</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\MDX Cube Provider</li> </ul>
GISConnectors =	<p>Location where the MicroStrategy GIS Connectors will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\GISConnectors</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\GISConnectors</li> </ul>
SystemManager =	<p>Location where the MicroStrategy System Manager will be installed. If left empty, it takes the default location of:</p> <ul style="list-style-type: none"> <li>32-bit Windows environments: C:\Program Files\MicroStrategy\SystemManager</li> <li>64-bit Windows environments: C:\Program Files (x86)\MicroStrategy\SystemManager</li> </ul>

## 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 =	<p>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.</p> <p>For an example of a response file used to uninstall all MicroStrategy products, see <a href="#">Uninstalling with a response.ini file, page 322</a>.</p>

## 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 Usher Security Administrator.
UserLastName =	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.
UserEmail =	Indicates the email of the currently logged user, or a user who is already registered. This email address is also used receive the badge invitation for your Usher Security network. If no information is provided, you cannot proceed.
CompanyName =	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.
LicenseKey =	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.

## Choose Destination Location dialog box

Options	Description
[SuiteTarget]	Section specifying the name of the target directory from where you can run the MicroStrategy products.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
TargetDirectory =	Location of the root directory for the Program Files. The default is set to: <ul style="list-style-type: none"> <li>32-bit Windows environment: C:\Program Files\MicroStrategy</li> <li>64-bit Windows environment: C:\Program Files (x86)\MicroStrategy</li> </ul>

## 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.

Options	Description
[ComponentSelection]	Equivalent to the Select Components dialog box that you see during installation. For the <code>Visible</code> option, you can either enter <code>TRUE</code> to show a product or <code>FALSE</code> to hide it. If you do not specify a <code>TRUE</code> or <code>FALSE</code> value for each product, <code>TRUE</code> is used for all products. For the <code>Select</code> option, you can enter <code>TRUE</code> to select the check box next to a product. If you enter <code>FALSE</code> , the check box next to the product is not selected.
<code>HideDialog =</code>	<code>TRUE</code> or <code>FALSE</code> . <code>FALSE</code> displays the dialog box. The default is <code>FALSE</code> .
<code>AnalyticsModulesVisible =</code>	MicroStrategy Analytics Modules
<code>ArchitectVisible =</code>	MicroStrategy Architect
<code>CommandManagerVisible =</code>	MicroStrategy Command Manager
<code>DeliveryEngineVisible =</code>	MicroStrategy Delivery Engine
<code>DeveloperVisible =</code>	MicroStrategy Developer
<code>AnalystVisible =</code>	MicroStrategy Analyst
<code>EnterpriseManagerVisible =</code>	MicroStrategy Enterprise Manager
<code>FunctionPluginVisible =</code>	MicroStrategy Function Plug-In Wizard
<code>GISConnectorsVisible =</code>	MicroStrategy GIS Connectors
<code>IntegrityManagerVisible =</code>	MicroStrategy Integrity Manager
<code>IServerDistributionServicesVisible =</code>	MicroStrategy Distribution Services
<code>IServerOLAPServicesVisible =</code>	MicroStrategy OLAP Services
<code>IServerReportServicesVisible =</code>	MicroStrategy Report Services
<code>IServerTransactionServicesVisible =</code>	MicroStrategy Transaction Services
<code>IServerVisible =</code>	MicroStrategy Intelligence Server
<code>RVisible =</code>	MicroStrategy Intelligence Server
<code>MobileServerASPVisible =</code>	MicroStrategy Mobile Server (ASP.NET)

Options	Description
MobileServerJSPVisible =	MicroStrategy Mobile Server (JSP)
MobileVisible =	MicroStrategy Mobile
NCSAdminVisible =	MicroStrategy Narrowcast Administrator
ObjectManagerVisible =	MicroStrategy Object Manager
OfficeVisible =	MicroStrategy Office
PortletsVisible =	MicroStrategy Portlets
SequeLinkVisible =	SequeLink ODBC Socket Server
ServerAdminVisible =	MicroStrategy Server Administrator
SubscriptionPortalVisible =	MicroStrategy Subscription Portal
SystemManagerVisible =	MicroStrategy System Manager
MDXCubeProviderVisible =	MicroStrategy MDX Cube Provider
TutorialDeliveryConfigureVisible =	MicroStrategy Tutorial - Delivery Configuration
TutorialDeliveryInstallVisible =	MicroStrategy Tutorial - Delivery Installation
WebAnalystVisible =	MicroStrategy Web Analyst
WebProfessionalVisible =	MicroStrategy Web Professional
WebReporterVisible =	MicroStrategy Web Reporter
WebServerASPNETVisible =	MicroStrategy Web Server (ASP.NET)
WebServerJSPVisible =	MicroStrategy Web Server (JSP)
WebServicesASPNETVisible =	MicroStrategy Web Services (ASP.NET)
WebServicesJSPVisible =	MicroStrategy Web Services (JSP)
UsherServerVisible =	Usher Security Server
UsherNetworkManagerVisible =	Usher Network Manager
UsherAnalyticsVisible =	Usher Analytics
UsherProfessionalVisible =	Usher Professional

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.

Options	Description
<code>AnalyticsModulesSelect =</code>	MicroStrategy Analytics Modules
<code>ArchitectSelect =</code>	MicroStrategy Architect
<code>CommandManagerSelect =</code>	MicroStrategy Command Manager
<code>DeliveryEngineSelect =</code>	MicroStrategy Delivery Engine
<code>DeveloperSelect =</code>	MicroStrategy Developer
<code>AnalystSelect =</code>	MicroStrategy Analyst
<code>EnterpriseManagerSelect =</code>	MicroStrategy Enterprise Manager
<code>FunctionPluginSelect =</code>	MicroStrategy Function Plug-In Wizard
<code>GISConnectorsSelect =</code>	MicroStrategy GIS Connectors
<code>IntegrityManagerSelect =</code>	MicroStrategy Integrity Manager
<code>IServerDistributionServicesSelect =</code>	MicroStrategy Distribution Services
<code>IServerOLAPServicesSelect =</code>	MicroStrategy OLAP Services
<code>IServerReportServicesSelect =</code>	MicroStrategy Report Services
<code>IServerSelect =</code>	MicroStrategy Intelligence Server
<code>IServerTransactionServicesSelect =</code>	MicroStrategy Transaction Services
<code>RSelect =</code>	R, R Integration Pack
<code>MobileSelect =</code>	MicroStrategy Mobile
<code>MobileServerASPSelect =</code>	MicroStrategy Mobile Server (ASP.NET)
<code>MobileServerJSPSelect =</code>	MicroStrategy Mobile Server (JSP)
<code>NCSAdminSelect =</code>	MicroStrategy Narrowcast Administrator
<code>ObjectManagerSelect =</code>	MicroStrategy Object Manager
<code>OfficeSelect =</code>	MicroStrategy Office
<code>PortletsSelect =</code>	MicroStrategy Portlets

Options	Description
SequeLinkSelect =	SequeLink ODBC Socket Server
ServerAdminSelect =	MicroStrategy Server Administrator
SubscriptionPortalSelect =	MicroStrategy Subscription Portal
SystemManagerSelect =	MicroStrategy System Manager
MDXCubeProviderSelect =	MicroStrategy MDX Cube Provider
TutorialDeliveryConfigureSelect =	MicroStrategy Tutorial - Delivery Configuration
TutorialDeliveryInstallSelect =	MicroStrategy Tutorial - Delivery Installation
WebAnalystSelect =	MicroStrategy Web Analyst
WebProfessionalSelect =	MicroStrategy Web Professional
WebReporterSelect =	MicroStrategy Web Reporter
WebServerASPNETSelect =	MicroStrategy Web Server (ASP.NET)
WebServerJSPSelect =	MicroStrategy Web Server JSP
WebServicesASPNETSelect =	MicroStrategy Web Services (ASP.NET)
WebServicesJSPSelect =	MicroStrategy Web Services (JSP)
UsherServerVisible =	Usher Security Server
UsherNetworkManagerVisible =	Usher Network Manager
UsherAnalyticsVisible =	Usher Analytics
UsherProfessionalVisible =	Usher Professional

## Installation Files

Options	Description
[IODSourceLocation]	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 <a href="#">Creating custom installation packages, page 77</a> .



Options	Description
Style =	<p>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:</p> <ul style="list-style-type: none"> <li><b>FILESERVER:</b> 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.</li> <li><b>HTTP:</b> 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.</li> <li><b>HTTPS:</b> 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.</li> </ul>
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 <code>URL</code> 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 <code>SourceLocation</code> 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

Options	Description
[UsherConfig]	Section that configures Usher Security Services.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
ExpressSkipUsherConfig=	TRUE or FALSE when <code>ExpressMode= TRUE</code> . TRUE to skip Usher configuration in Express Installation. Set to FALSE when <code>ExpressMode= FALSE</code> .

Options	Description
CACertificateChain =	Complete certificate chain for your SSL Server Certificate that you obtained from you IT Administrator. The path must be specified in an absolute format such as C:\folder\example.pem.
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.

### Open Source Software dialog box

Options	Description
[OpenSourceSoftwareDialog]	MySQL is requires as a repository in Express Install or required by Usher products in Custom Install.  R is used by Intelligence Server to process R based functions to enable R analytics
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE .
AgreeToDownloadOpenSourceSoftware=	TRUE or FALSE. TRUE to allow the installer to download the open source software components for you. FALSE to follow the links below to download the open source components yourself, being sure to

Options	Description
	<p>save all of the components to your Downloads folder. If not all the components are placed correctly in your Downloads folder, the Express Install or Custom Install cannot proceed.</p> <p>Please note that if you choose TRUE, you are authorizing the installer to download MySQL and R analytics components on your behalf. These MySQL and R analytics components are open-source software provided under the GPL licenses. These components are not provided by MicroStrategy. For access to the source code for these components, please visit <a href="http://www.mysql.com">http://www.mysql.com</a>, <a href="https://www.r-project.org/">https://www.r-project.org/</a>, and related links.</p> <p>MySQL:</p> <p><a href="http://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.28-winx64.zip">http://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.28-winx64.zip</a></p> <p>MySQL Connector/ODBC 5.3.4:</p> <p><a href="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</a></p> <p>MySQL Connector/Java 5.1.22:</p> <p><a href="http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.22.zip">http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.22.zip</a></p> <p>MySQL Connector/Python 2.1.3</p> <p><a href="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</a></p> <p>MySQL time zone description tables:</p> <p><a href="http://downloads.mysql.com/general/timezone_2015g_posix.zip">http://downloads.mysql.com/general/timezone_2015g_posix.zip</a></p>

## MicroStrategy Web virtual directory


Options	Description
[WebVirtualDirectory]	Section that specifies the virtual directory to be used for the MicroStrategy Web application.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
VirtualDirectory =	Enter a name for the virtual directory. The default is MicroStrategy.
RemoveVD =	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.

## MicroStrategy Mobile Server virtual directory

Options	Description
[MobileVirtualDirectory]	Section that specifies the virtual directory to be used for the MicroStrategy Mobile Server applications.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
VirtualDirectory =	Enter a name for the virtual directory. The default is MicroStrategyMobile.
ReconfigureVirtualDirectory =	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 <a href="#">Upgrade Guide</a> .
RemoveVD =	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.

## Health Center service account

Options	Description
[HealthCenterServiceAccount]	Section that configures the machine running the silent installation as a Health Agent or Master Health Agent.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
Port =	Type the port number to use to connect to the Health Agent machine. The default port is 44440.
AccessCode =	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.
SkipAccountSetting =	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.
Domain =	Type the domain of a Windows login with full administrative privileges under which to run the Health Center service.
Login =	Type the user name of a Windows login with full administrative privileges under which to run the Health Center service.

Options	Description
	 <p>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.</p>
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.
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 <i>Census</i> subfolder of the Health Center Repository. The default is FALSE.

### MicroStrategy MDX Cube Provider virtual directory

Options	Description
[MDXCubeProviderVirtualDirectory]	Section that specifies the virtual directory to be used for the MicroStrategy MDX Cube Provider.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
VirtualDirectory =	Enter a name for the virtual directory. The default is <i>MicroStrategyMDX</i> .
ReconfigureVirtualDirectory =	TRUE or FALSE. <b>Set this option to TRUE if the virtual directory for the MicroStrategy MDX Cube Provider should be reconfigured to support a new virtual directory.</b>
RemoveVD =	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.

## Subscription Portal virtual directory

Options	Description
[PortalVirtualDirectory]	Section that specifies the virtual directory to be used for MicroStrategy Subscription Portal.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
VirtualDirectory =	Enter a name for the virtual directory. The default is NarrowcastServer.
RemoveVD =	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.

## MicroStrategy Web Services virtual directory

Options	Description
[WebServicesDirectory]	Section that specifies the virtual directory to be used for MicroStrategy Web Services.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
VirtualDirectory =	Enter a name for the virtual directory. The default is MicroStrategyWS.
RemoveVD =	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.

## Intelligence Server service account

Options	Description
[IServerServiceAccount]	Section specifying the Windows account for the MicroStrategy Intelligence Server service. You have two options: <ul style="list-style-type: none"> <li>bypass entering the account information</li> <li>enter the account information</li> </ul>
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
SkipAccountSetting =	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

Options	Description
	is installing the products. The default is <code>FALSE</code> .
Domain =	Enter the domain where the account is located.
Login =	Enter the user name of the account to use.
Password =	Enter the password for the account.
ServiceStartup =	<code>AUTO</code> or <code>MANUAL</code> . Select to set the Intelligence Server service startup to be automatic or manual. The default is <code>AUTO</code> .

### Narrowcast Server service account

Options	Description
[NarrowcastServiceAccount]	Section specifying the Windows account from which the MicroStrategy Narrowcast Server service will run.
HideDialog =	<code>TRUE</code> or <code>FALSE</code> . <code>FALSE</code> displays the dialog box. The default is <code>FALSE</code> .
SkipAccountSetting =	<code>TRUE</code> or <code>FALSE</code> .  If you specify this value as <code>FALSE</code> , 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 <b>Next</b> to proceed with the installation process.
Domain =	Enter the domain where the account is located.
Login =	Enter the user name of the account to use.
Password =	Enter the password for the account.

### MicroStrategy Web Services and Web Services URL setting

Options	Description
[OfficeWebServicesURL]	Section specifying the URL for MicroStrategy Web Services and Web Services.
HideDialog =	<code>TRUE</code> or <code>FALSE</code> . <code>FALSE</code> displays the dialog box. The default is <code>FALSE</code> .
AllowBlankURL =	<code>TRUE</code> or <code>FALSE</code> . Specify whether to allow a blank URL. The installation routine validates the provided URL. If no URL is

Options	Description
	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 <code>TRUE</code> , the user message is not displayed if the URL is left blank. The default is <code>FALSE</code> .
URL =	Enter a URL pointing to a valid MicroStrategy Web Services installation, for example, <code>http://localhost/MicrostrategyWS/MSTRWS.asmx</code>

### MicroStrategy Office setting

Options	Description
[MSOfficeLoadOptions]	Section specifying the options that determine if the MicroStrategy Office toolbar is loaded in the installed Microsoft Office applications.
HideDialog =	<code>TRUE</code> or <code>FALSE</code> . <code>FALSE</code> displays the dialog box. The default is <code>FALSE</code> .
ConfigureExcel =	<code>TRUE</code> or <code>FALSE</code> . 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 <code>TRUE</code> .
ConfigureWord =	<code>TRUE</code> or <code>FALSE</code> . 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 <code>TRUE</code> .
ConfigurePowerpoint =	<code>TRUE</code> or <code>FALSE</code> . 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 <code>TRUE</code> .

### Intelligence Server definition setting

Options	Description
[ServerDefinitionSetting]	Section specifying whether MicroStrategy Intelligence Server will use the server definition included with the Tutorial.
HideDialog =	<code>TRUE</code> or <code>FALSE</code> . <code>FALSE</code> displays the dialog box. The default is <code>FALSE</code> .
OverwriteServerDefinition =	<code>TRUE</code> or <code>FALSE</code> . This option relates to the Tutorial. Set this option to overwrite existing MicroStrategy Intelligence Server definitions from a previous install. The default is <code>FALSE</code> .



## Analytics Module setting

Options	Description
[AnalyticsSetting]	Section that specifies the DSN used to connect to the MicroStrategy Analytics Modules.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.
OverwriteDSN =	<p>TRUE or FALSE. Set this option to overwrite an existing DSN with the same name. The data source names are as follows:</p> <ul style="list-style-type: none"> <li>Analytics_Metadata</li> <li>CAM_WH_AC</li> <li>FRAM_WH_AC</li> <li>HRAM_WH_AC</li> <li>MicroStrategy_Tutorial_Data</li> <li>SAM_WH_AC</li> <li>SDAM_WH_AC</li> </ul> <p>The default is FALSE.</p>

## Start Copying Files dialog box

Options	Description
[Summary]	Section that specifies the installation summary in the Start Copying Files dialog box.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.

## MicroStrategy Installation Wizard Complete dialog box

Options	Description
[Finish]	Section that specifies the MicroStrategy Installation Wizard Complete dialog box.
HideDialog =	TRUE or FALSE. FALSE displays the dialog box. The default is FALSE.

## 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
```

```
WebAnalystVisible=TRUE
WebProfessionalVisible=TRUE
WebReporterVisible=TRUE
WebServerASPNETVisible=TRUE
WebServerJSPVisible=TRUE
WebServicesASPNETVisible=TRUE
WebServicesJSPVisible=TRUE
OfficeVisible=TRUE
MobileVisible=TRUE
MobileClientVisible=TRUE
MobileServerASPVisible=TRUE
MobileServerJSPVisible=TRUE
AnalyticsModulesVisible=TRUE
NCSAdminVisible=TRUE
DeliveryEngineVisible=TRUE
SubscriptionPortalVisible=TRUE
TutorialDeliveryInstallVisible=TRUE
TutorialDeliveryConfigureVisible=TRUE
SequeLinkVisible=TRUE
PortletsVisible=TRUE
MDXCubeProviderVisible=TRUE
GISConnectorsVisible=TRUE
SystemManagerVisible=TRUE
UsherServerVisible=TRUE
UsherNetworkManagerVisible=TRUE
UsherAnalyticsVisible=TRUE
UsherProfessionalVisible=TRUE
RVisible=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
RSelect=TRUE
WebAnalystSelect=TRUE
WebProfessionalSelect=TRUE
WebReporterSelect=TRUE
WebServerASPNETSelect=TRUE
WebServerJSPSelect=TRUE
WebServicesASPNETSelect=TRUE
WebServicesJSPSelect=TRUE
OfficeSelect=TRUE
MobileSelect=TRUE
MobileClientSelect=TRUE
MobileServerASPSelect=TRUE
MobileServerJSPSelect=TRUE
AnalyticsModulesSelect=TRUE
NCSAdminSelect=TRUE
DeliveryEngineSelect=TRUE
SubscriptionPortalSelect=TRUE
TutorialDeliveryInstallSelect=TRUE
TutorialDeliveryConfigureSelect=TRUE
SequeLinkSelect=TRUE
PortletsSelect=TRUE
MDXCubeProviderSelect=TRUE
GISConnectorsSelect=TRUE
```

```
SystemManagerSelect=TRUE
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>
[OpenSourceSoftwareDialog]
HideDialog=TRUE
AgreeToDownloadOpenSourceSoftware=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
ConfigureWord=TRUE
[IServerServiceAccount]
HideDialog=TRUE
SkipAccountSetting=TRUE
Login=<NT_UserLoginHere>
Password=<UserPasswordHere>
Domain=<DomainHere>
ServiceStartUp=AUTO
[NarrowcastServiceAccount]
HideDialog=TRUE
SkipAccountSetting=TRUE
Login=<NT_UserLoginHere>
Password=<UserPasswordHere>
Domain=<DomainHere>
[Summary]
HideDialog=TRUE
[Finish]
HideDialog=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
[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>
[OpenSourceSoftwareDialog]
HideDialog=TRUE
AgreeToDownloadOpenSourceSoftware=TRUE
[Summary]
HideDialog=TRUE
[Finish]
HideDialog=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
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
MobileServerASPSelect=FALSE
MobileServerJSPSelect=FALSE
AnalyticsModulesSelect=FALSE
NCSAdminSelect=FALSE
DeliveryEngineSelect=FALSE
SubscriptionPortalSelect=FALSE
TutorialDeliveryInstallSelect=FALSE
TutorialDeliveryConfigureSelect=FALSE
SequeLinkSelect=FALSE
PortletsSelect=FALSE
MDXCubeProviderSelect=FALSE
GISConnectorsSelect=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
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
[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
[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
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
[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
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
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
[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
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) ###
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
[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 5, 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 188](#).

## Uninstalling with a response.ini file

You can uninstall all MicroStrategy products at once using a `response.ini` file. You must create a response file with the `RemoveAll` parameter set to `TRUE` in the Welcome section. This is also known as a silent uninstallation.



You must save the file as ANSI encoding.

Before uninstallation begins, the MicroStrategy application:

- Checks for user privileges. If they are not valid, uninstallation stops.
- Checks for running components. If one is found, uninstallation stops.
- Stops and deletes the MicroStrategy Intelligence Server service.
- Deletes application created files.


## Example of a response.ini file to uninstall MicroStrategy

You can use the following response file to remove all MicroStrategy products:

```
[Installer]
PropertiesFilesOverwrite=FALSE
EnableTracing=FALSE
HideAllDialogs=TRUE
ForceReboot=TRUE
PreventReboot=FALSE
CheckTCPIP=TRUE
CheckIIS=TRUE
CheckSP=TRUE
CreateShortcuts=TRUE
CheckRenameOperations=TRUE
AnalyticsOverwrite=FALSE
TutDeliveryOverwrite=FALSE
BackupFiles=FALSE
RunConfigWizard=FALSE
StopAllServices=TRUE
StopIIS=TRUE
EnableASPServices=FALSE
ConfigWizardResponseFile=
RegistrySizeReq=
LogFile=
[Welcome]
HideDialog=TRUE
RemoveAll=TRUE
[WebVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategy
RemoveVD=YES
[MobileVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyMobile
RemoveVD=YES
[OperationsManagerVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyOM
RemoveVD=YES
[PortalVirtualDirectory]
HideDialog=TRUE
VirtualDirectory=NarrowcastServer
RemoveVD=YES
[WebServicesDirectory]
HideDialog=TRUE
VirtualDirectory=MicroStrategyWS
RemoveVD=YES
[Finish]
HideDialog=TRUE
```

For details on creating a response.ini file, see [Creating a response.ini file, page 284](#).

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 354](#).

## 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.

Line	Description
Function [F]	Identifies the function calls in the setup script.
Information [I]	Logs information about the setup that is running.
Warning [W]	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 [S]	Includes fatal problems that prevent the setup from proceeding. For example, the Intelligence Server Service cannot be created and setup fails as a result.

A typical line in the `MSTRInst.log` file includes source file name, function name, source line number, and time. It appears as follows:

```
[Z:\InstallMaster\Setup\Script Files\MALicense.rul]
[UseDLL][l: 28][1318179ms][W] Le file C:\WINDOWS\TEMP\
{84D0D5E2-719A-11D7-847C-000C293B5019}\{B339F3B3-E78C-
45E8-B4D2-3C46C1C13663}\MAInst.dll couldn't be loaded
in memory.
```



When reviewing warning messages in the `MSTRInst.log` file, look for [W] and [S] to find related problems.

## Activating a silent installation

After the silent installation is complete, you must activate the MicroStrategy installation within 30 days. To activate the installation you can follow the instructions provided in [Chapter 5, Activating Your Installation](#).

## Troubleshooting silent installations

Silent installation may not work if you are installing in a different environment than the one you recorded. This is the case because any dialog box that was not recorded previously is not recognized, such as a prompt to stop your Intelligence Server or your IIS Web server. If this happens, verify:

- The version of Intelligence Server to ensure that you have the right one for the products you are installing
- 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 354](#). 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 327](#).

### 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 327](#).

- 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.



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 Access the disk drive through the command prompt.
- 5 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 356](#), 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 -fl"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 79* in *Chapter 1, Planning Your Installation*, for installation and configuration checklists.

# AUTOMATED INSTALLATION ON 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 Linux graphical user interface (GUI).



Intelligence Server configurations possible through the command line on Linux are covered in [Chapter 12, Configuring MicroStrategy Using Command Line Tools](#).

This chapter has one section: [Silent installation, page 330](#)

Before installing MicroStrategy products, see [Chapter 1, Planning Your Installation](#) for important preinstallation information.

## Silent installation

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

### Completing a silent installation

To run silent installation, you must create an options file and then run it with the MicroStrategy Installation Wizard. Save the options file as `options.txt`. An example of an `options.txt` file is provided in [Example of an options file, page 331](#) 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 333](#).

---

### 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_Linux` directory.
- 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 333](#).
- 6 Save the changes to the `options.txt` file.
- 7 To use the options file on a silent installation, specify `-silent -options FileName` as a command line argument to the wizard, where *FileName* is the name of this options file, for example, `options.txt`. For example, type the command:

```
setup.sh -silent -options options.txt
```

If you are installing a MicroStrategy hotfix, use the `setupHF.sh` file instead. For example, type the command:

```
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 333](#).



Note 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 `.tzip` 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 347](#).

```
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=UserPassword
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 115](#).

- 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:

- A visible option, which can use one of the following values:
  - `true`: Indicates that the feature is displayed in the MicroStrategy Installation Wizard as available for installation.
  - `false`: Indicates that the feature is not displayed in the MicroStrategy Installation Wizard as available for installation. If you define a product's visible option as false, it cannot be installed.

If no value is specified, the default is `true` for all products. You can also define the visible option for all products using the parameter `SelectComponents.visible`. You can exclude these visible options for each product if you are using the options file as part of a completely silent installation where no user interface is displayed to the user.

- An active option, which can use one of the following values:
  - `true`: Indicates that the feature is selected for installation.
  - `false`: Indicates that the feature is not selected for installation, or the product is to be uninstalled as part of an installation that modifies or uninstalls previously installed MicroStrategy software.

To review a description of each MicroStrategy project, see [MicroStrategy products and components, page 19](#).

## 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 336](#)
- [MicroStrategy OLAP Services, page 336](#)
- [MicroStrategy Distribution Services, page 337](#)
- [MicroStrategy Transaction Services, page 337](#)

## 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:

```
WebServices.visible=Value
```

The selection state of MicroStrategy Web Services J2EE.

```
WebServices.active=Value
```

For example, to select MicroStrategy Web Services J2EE for installation, use:

```
WebServices.visible=true
```

```
WebServices.active=true
```

### **MicroStrategy Mobile Server JSP**

The state of whether MicroStrategy Mobile Server JSP is displayed in the MicroStrategy Installation Wizard:

```
MobileServer.visible=Value
```

The selection state of Mobile Server JSP.

```
MobileServer.active=Value
```

For example, to select MicroStrategy Mobile Server JSP for installation, use:

```
MobileServer.visible=true
```

```
MobileServer.active=true
```

## MicroStrategy Command Manager

The state of whether MicroStrategy Command Manager is displayed in the MicroStrategy Installation Wizard:

```
CommandManager.visible=Value
```

The selection state of MicroStrategy Command Manager.

```
CommandManager.active=Value
```

For example, to select MicroStrategy Command Manager for installation, use:

```
CommandManager.visible=true
```

```
CommandManager.active=true
```

## MicroStrategy System Manager

The state of whether MicroStrategy System Manager is displayed in the MicroStrategy Installation Wizard:

```
SystemManager.visible=Value
```

The selection state of MicroStrategy System Manager.

```
SystemManager.active=Value
```

For example, to select MicroStrategy System Manager for installation, use:

```
SystemManager.visible=true
```

```
SystemManager.active=true
```

## MicroStrategy Integrity Manager

The state of whether MicroStrategy Integrity Manager is displayed in the MicroStrategy Installation Wizard:

```
IntegrityManager.visible=Value
```

The selection state of MicroStrategy Integrity Manager.

```
IntegrityManager.active=Value
```

For example, to select MicroStrategy Integrity Manager for installation, use:

```
IntegrityManager.visible=true
```

```
IntegrityManager.active=true
```

### **MicroStrategy Enterprise Manager**

The state of whether MicroStrategy Enterprise Manager is displayed in the MicroStrategy Installation Wizard:

```
EnterpriseManager.visible=Value
```

The selection state of MicroStrategy Enterprise Manager.

```
EnterpriseManager.active=Value
```

For example, to select MicroStrategy Enterprise Manager for installation, use:

```
EnterpriseManager.visible=true
```

```
EnterpriseManager.active=true
```

### **Usher Security Server**

The state of whether Usher Security Server is displayed in the MicroStrategy Installation Wizard:

```
UsherSecurityServer.visible=Value
```

The selection state of Usher Security Server.

```
UsherSecurityServer.active=Value
```

For example, to select Usher Security Server for installation, use:

```
UsherSecurityServer.visible=true
```

```
UsherSecurityServer.active=true
```

### **Usher Network Manager**

The state of whether Usher Network Manager is displayed in the MicroStrategy Installation Wizard:

```
UsherNetworkManager.visible=Value
```

The selection state of Usher Network Manager.

```
UsherNetworkManager.active=Value
```

For example, to select Usher Network Manager for installation, use:

```
UsherNetworkManager.visible=true
```

```
UsherNetworkManager.active=true
```

### **Usher Analytics**

The state of whether Usher Analytics is displayed in the MicroStrategy Installation Wizard:

```
UsherAnalytics.visible=Value
```

The selection state of Usher Analytics.

```
UsherAnalytics.active=Value
```

For example, to select Usher Analytics for installation, use:

```
UsherAnalytics.visible=true
```

```
UsherAnalytics.active=true
```

## Usher Professional

The state of whether Usher Professional is displayed in the MicroStrategy Installation Wizard:

```
UsherMobile.visible=Value
```

The selection state of Usher Professional.

```
UsherMobile.active=Value
```

For example, to select Usher Professional for installation, use:

```
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.

```
cpuCount.number=Value
```

By default, the maximum number of CPUs is allowed. This is represented with the following value for this parameter:

```
cpuCount.number=maximum
```

## MicroStrategy product and component installation locations

You can define the installation locations for the following products and components:

- MicroStrategy Web Universal Install Location:

```
WebUniversal.deployDirectory=Value
```

- MicroStrategy Portlets Install Location:

```
Portlets.installDirectory=Value
```

- MicroStrategy GIS Connectors Install Location

`GISConnectors.installDirectory=Value`

- MicroStrategy 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 Professional 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 `.tzip` 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 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 77](#). Once you determine and collect the `.tzip` 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 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 125](#) in [Chapter 3, Installing MicroStrategy on Linux](#).

## Silent installation output

The installation returns 0 if the installation is successful, and any other value if it is not. The `install.log` file in the `InstallPath` directory provides more information on possible errors. For more information on the `install.log` file, see [Installation log file, page 282](#) in [Chapter 2, Installing MicroStrategy on Windows](#).



If the installation fails on any of the steps before it starts copying the files, it does not give any feedback other than the return value different from 0.

## Activating a silent installation

After the silent installation is complete, you must activate the MicroStrategy installation within 30 days. To activate the installation you can follow the instructions provided in [Chapter 5, Activating Your Installation](#).

## Configuring MicroStrategy in command line mode

The MicroStrategy Configuration Wizard is provided in command line mode so that you can use the Configuration Wizard through the operating system console if you do not have access to the GUI. Running the Configuration Wizard in command line mode to configure MicroStrategy on Linux machines is covered in the [Configuring MicroStrategy with a response.ini file, page 364](#) section in [Chapter 12, 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, see [Installing and configuring MicroStrategy on Linux, page 80](#) in [Chapter 1, Planning Your Installation](#), for an installation and configuration checklist.

# DEPLOYING OEM APPLICATIONS

This chapter explains the common workflow for deploying the MicroStrategy platform as an Original Equipment Manufacturer (OEM) application.

The MicroStrategy platform can be deployed as an OEM application in various ways:

- MicroStrategy can be deployed as a software as a service model through the use of MicroStrategy Web. In this scenario MicroStrategy is installed and configured at a centralized location using the standard process, and the customized application is deployed as an OEM application using MicroStrategy Web. For information on deploying MicroStrategy Web, see [Chapter 7, Deploying MicroStrategy Web and Mobile Server](#). For information on customizing MicroStrategy Web, see [Customizing MicroStrategy Web, page 353](#).
- MicroStrategy can be deployed as part of an OEM software bundle directly to a customer environment. This chapter focuses on the development and deployment of this type of OEM application.

The following table lists a best practices checklist of how to deploy MicroStrategy as an OEM application. In addition to using this checklist, you can use MicroStrategy System Manager. System Manager allows you to define multiple configurations for your MicroStrategy environment that can be executed in a single workflow. For steps to use System Manager to deploy MicroStrategy configurations, see the [System Administration Guide](#).

Complete	Task
	Install MicroStrategy on an OEM environment. For installation information, see: <ul style="list-style-type: none"> <li>• <a href="#">Chapter 2, Installing MicroStrategy on Windows</a></li> <li>• <a href="#">Chapter 3, Installing MicroStrategy on Linux</a></li> </ul>
	Create DSNs using the Connectivity Wizard, as described in <a href="#">Creating DSNs for OEM environments, page 351</a> .
	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 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

Complete	Task
	<a href="#">Configuring a MicroStrategy installation, page 352.</a>
	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 <a href="#">Designing a project and reporting environment, page 352.</a>
	Customize MicroStrategy Web through the use of the MicroStrategy SDK, as described in <a href="#">Customizing MicroStrategy Web, page 353.</a>
	Deploy a MicroStrategy OEM application on an OEM's customer environment, as described in <a href="#">Deploying a MicroStrategy OEM application, page 353.</a>
	Create DSNs on the OEM's customer environment as necessary, as described in <a href="#">Creating DSNs for OEM environments, page 351.</a>
	Configure and tune an OEM deployment through the use of various MicroStrategy tools, as described in <a href="#">Tuning an OEM deployment, page 358.</a>
	If you are modifying a project that has already been deployed as an OEM application, see <a href="#">Updating OEM applications, page 359</a> for best practices on how to incorporate any custom reports or objects that may have been created for the deployed application.
	Troubleshoot your MicroStrategy OEM applications using MicroStrategy Health Center, as described in <a href="#">Troubleshooting support for MicroStrategy OEM applications, page 360.</a>

## 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 157.](#)

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 352](#).

A DSN can be created using the MicroStrategy Connectivity Wizard, as described in [Defining DSNs, page 161](#).

## Creating DSNs as part of an OEM deployment

As part of the deployment of an OEM application, the Connectivity Wizard can be run from the command line to create DSNs on 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 361](#).

For OEM deployments on Windows machines, use the Connectivity Wizard interface to create DSNs, as described in [Defining DSNs, page 161](#).

## 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 166](#).

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 188](#).

## 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 359](#) 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>. Alternately, you can access the MicroStrategy Developer Library from the MicroStrategy support site.

## Deploying a MicroStrategy OEM application

After an OEM application is developed, it then must be deployed to the customer's environment. Steps to deploy an 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 354](#). To use the Installation Wizard to install MicroStrategy products, see:

- [Chapter 2, Installing MicroStrategy on Windows](#)

- [Chapter 3, Installing MicroStrategy on Linux](#)
- 2 Additional configurations are required, as described in [Configuring an OEM deployment installation, page 357](#).

## OEM silent installations

You can use silent installation to easily embed MicroStrategy products with other applications. The steps below show you how to use a silent installation to deploy an OEM application on a Windows environment. For additional information on silent installations, see [Chapter 9, Automated Installation on Windows](#).

You can follow the steps below to perform a silent installation on a Windows environment. To perform a silent installation on a Linux environment, see [Silent installation, page 330](#).

### 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 283](#).

Response File Section	Selection
[Installer]	Required
HideAllDialogs =	Required
PreventReboot =	Optional
StopAllServices =	Optional
StopIIS =	Optional
CheckRenameOperations =	Optional
[UserRegistration]	Required
[ComponentSelection]	Required

Response File Section	Selection
EnterpriseManagerSelect =	Required
[InitialPaths]	Required
EnterpriseManager =	Required

 Setting `HideAllDialogs = TRUE` causes the script for the response file to:

- Use default values as specified in the `response.ini` file.
- Not require user input.
- Keep the dialog flow consistent from one instance to the next. Consistency in the response file script from one instance to the next is necessary; if `setup.iss` detects an inconsistency in the dialog flow, installation is terminated and a log file for the failure is created.

The only dialog flow modifications pertinent to silent installation are specific to file location. Therefore, the only portion of the `response.ini` that may need to be modified is the `[InitialPaths]` section.

The rest of this procedure assumes you have saved the `response.ini` file to the file path `C:\`. If you save it to another file path, replace `C:\response.ini` with the file path of your `response.ini` file.

You must save the `response.ini` file as ANSI encoding.

- 2 Create the `setup.iss` file to use in conjunction with the `response.ini` file for the silent installation. Use a text editor to create the `setup.iss` file with the following information:

```
[InstallShield Silent]
Version=v7.00
File=ResponseFile
[File Transfer]
OverwrittenReadOnly=NoToAll
[Application]
Name=MicroStrategy
Version=x.y.z
#x.y.z represent the version of the
product#
Company=MicroStrategy
Lang=LanguageValue
[{8CCF3F6C-55B7-4A27-8C68-ADF21D0585A2}-DlgOrder]
Count=0
```

You must save the `setup.iss` file as ANSI encoding.



The version in the `setup.iss` file must match the MicroStrategy version you are installing exactly. For example, if you are installing version 10.7.0 you must enter `Version=10.7.0`. Entering a version as `Version=10` or `Version=10.7.x` causes an error when trying to perform a silent installation of version 9.3.0.

For an explanation of the `LanguageValue` parameter within the line `Lang=LanguageValue`, see [Language settings for silent installations, page 356](#).

- 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:



For an explanation of the `LanguageValue` parameter, see [Language settings for silent installations, page 356](#)

```
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.

Language	Value
Danish	0006
Dutch	0019
English	0009
French	0012
German	0007
Italian	0016
Japanese	0017
Korean	0018
Portuguese	0022
Simplified Chinese	2052
Spanish	0010
Swedish	0029
Traditional Chinese	1028

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 351](#).
- 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 352](#).
- Configure and tune an OEM deployment through the use of various MicroStrategy tools, as described in [Tuning an OEM deployment, page 358](#).

## 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 Linux machines is covered in [Configuring MicroStrategy with a response.ini file, page 364](#).

## 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 359](#).
- 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 360](#).

### 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](#).



# CONFIGURING MICROSTRATEGY USING COMMAND LINE TOOLS

MicroStrategy tools are provided in command line mode on Linux so that you can perform various configuration tasks through the operating system console. This enables you to perform your required configurations even if you do not have access to the MicroStrategy interface.

On a Windows machine, it is recommended to use the appropriate MicroStrategy interfaces to perform the configurations described in this chapter.



When you perform MicroStrategy configuration tasks through a Linux operating system console, you must make sure reserved words and characters are not mistakenly included in your commands. To avoid issues caused by reserved words and characters, see [Supporting reserved words and characters, page 382](#).

This chapter covers the configurations listed below:

- [Creating a DSN for a data source, page 361](#)
- [Testing ODBC connectivity](#)
- [Configuring MicroStrategy with a response.ini file, page 364](#)
- [Configuring and controlling Intelligence Server, page 378](#)

## Creating a DSN for a data source

After you install an ODBC driver (see [Appendix A, Connecting to Databases and Data Sources](#)), you can define one or more data sources for it. The DSN should provide a unique description of the data, for example, `Payroll_Project_Metadata` or `Payroll_Warehouse`.

The DSN is the name for a pointer used by a client application (in this case MicroStrategy) to find and connect to a data source. Multiple DSNs can point to the same data source and one DSN can be used by different applications.

MicroStrategy provides a one-line command line version of the MicroStrategy Connectivity Wizard to create DSNs on Linux.

 You can create DSNs using the MicroStrategy Connectivity Wizard on Windows and Linux machines, as described in [Communicating with databases, page 157](#).

---

## To create a DSN on Linux from the command line

---


- 1 From a Linux console window, browse to *HOME\_PATH*, where *HOME\_PATH* is the directory that you specified as the home directory during installation.
- 2 Browse to the folder `bin`.
- 3 Type `mstrconnectwiz -h`, and then 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:OracleUserPassword
```

## Testing ODBC connectivity

ODBC connectivity is one of two layers of connectivity that are listed in the next table, along with the associated connectivity testing programs. Connectivity should be tested from the bottom up—the network layer first and then the ODBC layer.

Layer	Test with
ODBC driver	Test ODBC <code>mstrtestodbc</code> or <code>mstrtodbx</code>
Network TCP/IP	Simple Network Layer Testing Tool Ping, <code>PING.EXE</code> , for TCP/IP

 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 12, Configuring MicroStrategy Using Command Line Tools*.
  - On Linux using the DB Query Tool interface, perform the following steps:
    - In a Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory that you specified as the home directory during installation.
    - Browse to the folder `bin` and type `./mstrdbquerytool`, then press `ENTER`.
  - On Linux from the command line, perform the following steps:
    - In a Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory that you specified as the home directory during installation.
    - Browse to the folder `bin` and type `./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 12, Configuring MicroStrategy Using Command Line Tools*.
- 2 From the **Session** menu, select **Open Connection**, or click the **Connect** icon on the toolbar. The Connect dialog box opens. The connection interface varies depending on the destination database.
- 3 Select the DSN for a data source.
- 4 Enter the appropriate user name and password.
- 5 Click **Connect**. After your connection is opened, the connection string is displayed in the MicroStrategy DB Query Tool at the bottom. Your cursor is inserted automatically in the SQL Statement window.
- 6 In the SQL Statement window, type a SQL query such as:

```
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 machine using a `response.ini` file:

- [Creating a response.ini file, page 365](#)
- [Using the response.ini file to configure MicroStrategy, page 378](#)
- [Parameters and options in the response.ini file, page 378](#)



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 188](#).

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 166](#).

## Creating a response.ini file

This section describes how to configure MicroStrategy using the command line mode. Performing the steps in this section creates a `response.ini` file that can be used to configure MicroStrategy installations on Linux machines.

---

### To configure MicroStrategy using the command line mode

---

- 1 In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.
- 2 Browse to the `bin` directory.
- 3 At the command prompt, type `mstrcfgwiz-editor`, then 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 378](#).
  - 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 365](#) 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 366](#) 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 373](#) 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 376](#) for steps to create project sources.

## Creating metadata, History List, and statistics tables

If you selected option 1 in [Configuration tasks, page 365](#), 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 366](#)
- [Creating History List tables, page 368](#)
- [Creating statistics tables, page 371](#)

### 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 365](#).

---

### 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 Linux machine, see [Creating a DSN for a data source, page 361](#).

- 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 367](#) 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 367](#) 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 371](#).
- 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 368](#) 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 371](#).
- 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 368](#) 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 378](#).

- 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 378](#).

### 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 365](#).



## To create History List tables


- 1 After you create metadata tables (see [Creating metadata tables, page 366](#)) 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 369](#) 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 370](#) 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 369](#) 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`.

 If you do not have a DSN defined on your Linux machine, see [Creating a DSN for a data source, page 361](#).

- 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`.
- 5 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 370](#) 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 370](#) 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 History List script**

- 8** 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**

- 9** 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 371](#).
- 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 370](#) 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 378](#).

- 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 378](#).

## 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 365](#).

## To create statistics tables

- 1** After you create metadata tables (see [Creating metadata tables, page 366](#)), create History List tables ([Creating History List tables, page 368](#)), 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 372](#) 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 372](#) 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 372](#) 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 372](#) 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 372](#) 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 378](#).
  - 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 378](#).

## Setting up MicroStrategy Intelligence Server

If you selected option 2 in [Configuration tasks, page 365](#), 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.
- 4 You can enter characters to use as a prefix for the names of your metadata tables or use no prefix, as described below:
  - Type the required prefix characters, and then 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.
- 5 You can enter characters to use as a prefix for the names of temp tables or use no prefix, as described below:
  - Type the required prefix characters, and then 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.

- 6 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 you initial configuration.

- 7 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.
- 8 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 374](#) 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 374](#).
  - Type 3, and then press `ENTER` to delete a server definition. Refer to [Deleting a server definition, page 376](#) 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 374](#).

## 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 373](#).

## 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 373](#).

## 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 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.
- 8 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 378](#).

- 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 378](#).

## 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 373](#).

---

## 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 378](#).
  - 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 378](#).

## Creating a project source

If you selected option 3 in [Configuration tasks, page 365](#), 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 187](#).

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 378](#).
  - 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 378](#).

## 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 365](#).

---

### To use the response.ini file through the Configuration Wizard in command line mode

---

- 1 In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.
- 2 Browse to the 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 191](#).

## Configuring and controlling Intelligence Server

MicroStrategy provides various command line tools to configure and control Intelligence Servers running on Linux. Each command line tool provides descriptive prompts and help information to guide you on how you can use the tool. This section gives a general overview of each tools functionality, and how to access more detailed information on how to use the tools.



On a Windows machine, these configurations can be completed with MicroStrategy Developer, Service Manager, and other MicroStrategy tools. For information on performing various administrative tasks, see the [System Administration Guide](#).

## Starting, configuring, and monitoring Intelligence Server with mstrsvr

If your Intelligence Server is installed on a Linux machine, you can start, configure, and monitor your Intelligence Server from the command line with `mstrsvr`. This tool starts

Intelligence Server from the command line and displays the following information about your Intelligence Server connection:

- Intelligence Server version number
- Intelligence Server instance name
- Metadata DSN
- Metadata login
- Intelligence Server definition name
- Port number

You can then perform various configuration and monitoring tasks for your running Intelligence Server, which includes but is not limited to:

- Display database connection information
- Open, idle, and resume projects
- Check and close jobs
- Monitor users
- Define server clustering options
- Monitor memory usage information
- Stop the server
- Monitor lock contentions

---

## To start, configure, and monitor Intelligence Server with `mstrsvr`

---

- 1 From a Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory that you specified as the home directory during installation.
- 2 Browse to the folder `bin`.
- 3 Type `mstrsvr`, 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 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 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 381](#).

You do not need to enter any command to quit the `mstrctl` tool because it is a one-line command line tool.

## Using files to store output and provide input

With the `mstrctl` command line tool, you can perform the following tasks:

- Display and modify a server configuration
- Display and modify a service configuration
- Display and modify a server instance configuration

The commands that display the configurations listed above output long XML definitions to the command line. The commands that modify the configurations listed above require a long XML definition as input.

Rather than displaying and inputting long XML definitions from the command line, you can use files to store and provide input for long XML definitions.



- Configuring Intelligence Server with XML files requires extensive knowledge of the various parameters and values used to define Intelligence Server configurations. Providing an incorrect XML definition to configure Intelligence Server can cause errors and unexpected functionality.
- Prior to using commands to display and modify service configurations (`gsvc` and `ssvc`) you must register Intelligence Server as a service. You can perform this task by using the `rs` command for `mstrctl`. To register an Intelligence Server as a service on a Linux machine, you must be logged in with an account that has root user privileges and permissions.

The following commands can have their output sent to a file:

- `gsc`: Displays a server configuration
- `gsvc`: Displays a service configuration
- `gsic`: Displays a server instance configuration

For example, you can run the following command to output the server instance configuration to an XML file:

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

A `ServerInstance.xml` file is saved in the current directory.

The following commands can read input from a file:

- `ssc`: Modifies a server configuration
- `ssvc`: Modifies a service configuration
- `ssic`: Modifies a server instance configuration

For example, you can run the following command to modify the server instance configuration by reading input from an XML file:

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

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

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

## Supporting reserved words and characters

When you perform MicroStrategy configuration tasks through the Linux operating system console, you must make sure reserved words and characters are not mistakenly included in your commands.

Linux operating system consoles use reserved words and characters to perform various actions. For example, the `$` character may perform an action when included as part of a command executed through the operating system console. If this character is included in a command to configure, it can cause the command to fail.

For example, you use the following command to create a DSN to an Oracle database:

```
mstrconnectwiz ORCLW $MyOracleDSN 12.34.56.78 orcl 1521 -
u:OracleUser -p:OracleUserPassword
```

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:OracleUserPassword
```

This time, the name of the DSN `$MyOracleDSN` is enclosed by single quotes, which allows all of the characters to be interpreted as a string of characters.

## Configuring your MicroStrategy installation

To help guide the rest of your installation and configuration steps, refer to the section [Installing and configuring MicroStrategy on Linux, page 80](#) in [Chapter 1, Planning Your Installation](#), for an installation and configuration checklist.

# ADDING OR REMOVING MICROSTRATEGY COMPONENTS

This chapter describes how to add or remove MicroStrategy components on different operating systems.

This chapter includes the following sections:

<a href="#">Adding or removing MicroStrategy components on Windows</a>	383
<a href="#">Re-installing MicroStrategy components on Windows</a>	384
<a href="#">Uninstalling MicroStrategy components on Windows</a>	385
<a href="#">Uninstalling MicroStrategy components on Linux</a>	387

## Adding or removing MicroStrategy components on Windows

You 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 MicroStrategy products.
- 2 Open the Microsoft Control Panel and navigate to the options to add or remove programs. See the Microsoft documentation for steps to access these options.
- 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 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 386](#) and [Uninstalling MicroStrategy Health Center, page 387](#).

For more details on each page of the MicroStrategy Installation Wizard, see [Chapter 2, Installing MicroStrategy on Windows](#).

## Re-installing MicroStrategy components on Windows

You can re-install MicroStrategy components if there are problems with previously installed components. During re-installation the list of components previously installed are displayed and these components are re-installed. If you installed the MicroStrategy components using a disk, you need your original installation disk to repair the installation.



The re-installation of MicroStrategy Office must be performed separately. The procedure for re-installing MicroStrategy Office is explained in the following section.

---

### To re-install MicroStrategy components

---

- 1 Close all 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 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 MicroStrategy products.
- 2 Open the Microsoft Control Panel and navigate to the options to add or remove programs. See the appropriate Microsoft documentation for steps to access these options.
- 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 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

---

- 1 Close all MicroStrategy products.
- 2 Open the Microsoft Control Panel and navigate to the options to add or remove programs. See the appropriate Microsoft documentation for steps to access these options.
- 3 Within the list of installed programs, select **MicroStrategy** and click **Change**. The MicroStrategy Setup Maintenance program opens.
- 4 Select **Remove** and click **Next**.
- 5 Click **Yes** to any prompts that appear.
- 6 If prompted to stop your Web server, click **Yes** to stop it and continue with the uninstallation.
- 7 After the uninstall 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

---

- 1 Close all MicroStrategy products.

- 2 Open the Microsoft Control Panel and navigate to the options to add or remove programs. See the appropriate Microsoft documentation for steps to access these options.
- 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 When prompted, select **Yes** to continue with the uninstall procedure.
- 6 When the uninstall is complete, click **Finish**.

## 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

---

- 1 Close all MicroStrategy products.
- 2 Open the Microsoft Control Panel and navigate to the options to add or remove programs. See the appropriate Microsoft documentation for steps to access these options.
- 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 is complete, click **Finish** to close the maintenance program.

## Uninstalling MicroStrategy components on Linux

This section discusses how to uninstall MicroStrategy Intelligence Server and other MicroStrategy products on a Linux platform.

---

### To uninstall MicroStrategy products on Linux

---

- 1 In a Linux console window, browse to `INSTALL_PATH`, where `INSTALL_PATH` is the directory you specified as the install directory during installation.
- 2 Browse to the `/_uninst` folder.
- 3 You can uninstall MicroStrategy products using the MicroStrategy Installation Wizard or through the command line as described in the options below:

- To uninstall using the MicroStrategy Installation Wizard, type `./setup.sh` and press **ENTER**. Follow the steps described in [To complete uninstallation with the MicroStrategy Installation Wizard, page 388](#).
- 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 prompts for a language. Specify the language to be used for the MicroStrategy uninstallation and click **OK** to proceed to the next step.
- 5 The MicroStrategy Installation Wizard opens with the Welcome page displayed. Click **Next**. The MicroStrategy Installation Selection page opens.
- 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 388](#) 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 388](#) 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 Linux, you can also configure parameters with the `odbc.ini` file. This appendix discusses the following topics:

- [Creating DSNs for specific data sources, page 392](#): Configuration information required to create a DSN for data sources available through the Connectivity Wizard.
- [Creating database connections in Web, page 426](#): 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 428](#): Configuration information required to configure data sources using the `odbc.ini` file in a Linux environment.

## Creating DSNs for specific data sources

You can create a DSN for data sources available through the Connectivity Wizard. The following table lists the information required for each type of data source when you create a new DSN using the Connectivity Wizard. For information on what operating systems each ODBC driver is certified for, see [Certified ODBC drivers for MicroStrategy Intelligence Server, page 72](#).

- For ODBC-specific driver details, refer to the different ODBC driver sections below the table.
- You can create a DSN from the command line version of the Connectivity Wizard in Linux. Browse to `HOME_PATH/bin`, where `HOME_PATH` is the directory you specified as the home directory during installation. In the console window, type `./mstrconnectwiz -h`, then press ENTER. This command displays command line syntax and examples for different database platforms. Create your command based on the syntax and examples displayed. Once you run your command, a DSN is created in the `odbc.ini` file.



	Driver Details
<i>MicroStrategy ODBC Driver for Apache Hive Wire Protocol for Windows and Linux, page 395</i>	<ul style="list-style-type: none"> <li>• Data source name</li> <li>• Host name</li> <li>• Database name</li> <li>• Port number (10000 in most cases)</li> </ul>
<i>MicroStrategy ODBC Driver for Amazon Redshift Wire Protocol for Windows and Linux, page 396</i>	<ul style="list-style-type: none"> <li>• Data source name</li> <li>• Host name</li> <li>• Port number (5439 in most cases)</li> <li>• Database name</li> </ul>
<i>MicroStrategy ODBC Driver for DB2 Wire Protocol for Windows and Linux, page 396</i>	<ul style="list-style-type: none"> <li>• Data source name</li> <li>• Host name</li> <li>• Database name</li> <li>• Port number</li> </ul>
<i>MicroStrategy ODBC Driver for DB2 Wire Protocol for iSeries for Windows and Linux, page 397</i>	<ul style="list-style-type: none"> <li>• Data source name</li> <li>• IP address</li> <li>• Collection</li> <li>• Location</li> <li>• Isolation level</li> <li>• Package owner</li> <li>• TCP port (446 in most cases)</li> </ul>
<i>MicroStrategy ODBC Driver for DB2 z/OS for Windows and Linux, page 397</i>	<ul style="list-style-type: none"> <li>• Data source name</li> <li>• IP address</li> <li>• Collection</li> <li>• Location</li> <li>• Package collection</li> <li>• Package owner</li> <li>• TCP port (446 in most cases)</li> </ul>
<i>MicroStrategy ODBC Driver for Impala Wire Protocol for Windows and Linux, page 398</i>	<ul style="list-style-type: none"> <li>• Data source name</li> <li>• Host name</li> <li>• Port number (21050 in most cases)</li> <li>• Database name</li> </ul>
<i>MicroStrategy ODBC Driver for Informix Wire Protocol for Windows and Linux, page 398</i>	<ul style="list-style-type: none"> <li>• Data source name</li> <li>• Server name</li> <li>• Host name</li> </ul>

	Driver Details
	<ul style="list-style-type: none"> <li>Port number (1526 in most cases)</li> <li>Database name</li> </ul>
<i>MicroStrategy ODBC Driver for Microsoft SQL Server for Windows and Linux, page 398</i>	<ul style="list-style-type: none"> <li>Data source name</li> <li>Host name</li> <li>Port number</li> <li>Database name</li> <li>Use Windows NT Authentication for login ID</li> <li>Enable SQL Database (Azure) support</li> </ul>
<i>MicroStrategy ODBC Driver for MongoDB for Windows and Linux, page 400</i>	<ul style="list-style-type: none"> <li>Data source name</li> <li>Host name</li> <li>Port number</li> <li>Database name</li> </ul>
<i>MicroStrategy ODBC Driver for Oracle Wire Protocol for Windows and Linux, page 401</i>	<p>Data source name and either:</p> <p>Standard connection:</p> <ul style="list-style-type: none"> <li>Host name</li> <li>Port number (in most cases, 1521)</li> <li>SID (MicroStrategy default is ORCL)</li> <li>Service name</li> <li>Alternate servers</li> </ul> <p>TNSNames connection:</p> <ul style="list-style-type: none"> <li>Server name</li> <li>TNSNames file</li> </ul>
<i>MicroStrategy ODBC Driver for Pivotal Greenplum Wire Protocol for Windows and Linux, page 402</i>	<ul style="list-style-type: none"> <li>Data source name</li> <li>Host name</li> <li>Port number (5432 in most cases)</li> <li>Database name</li> </ul>
<i>MicroStrategy ODBC Driver for PostgreSQL Wire Protocol for Windows and Linux, page 403</i>	<ul style="list-style-type: none"> <li>Data Source Name</li> <li>Host Name</li> <li>Port Number</li> <li>Database Name</li> <li>User Name</li> </ul>
<i>MicroStrategy ODBC Driver for SAP Sybase ASE Wire Protocol</i>	<ul style="list-style-type: none"> <li>Data source name</li> <li>Network address</li> </ul>

	Driver Details
<i>for Windows and Linux, page 403</i>	<ul style="list-style-type: none"> <li>Database name</li> <li>Enable unicode support</li> </ul>
<i>MicroStrategy ODBC Driver for SequeLink, page 403</i>	The MicroStrategy ODBC Driver for SequeLink allows you to access Microsoft Access databases or Microsoft Excel files stored on a Windows machine from an Intelligence Server hosted on a Linux machine.
<i>MicroStrategy ODBC Driver for Salesforce, page 407</i>	The MicroStrategy ODBC Driver for Salesforce allows you to access resources on Salesforce.com, from an Intelligence Server hosted on a Linux machine.

This section also provides information on how to install and configure drivers from other vendors with MicroStrategy:

- *ODBC Driver for Red Brick for Linux, page 408*
- *ODBC Driver for Sybase Adaptive Server IQ for Linux, page 409*
- *ODBC Driver for Teradata for Linux, page 410*
- *ODBC Driver for Informix 8 for Linux, page 411*
- *ODBC Driver for Netezza for Linux, page 412*
- *ODBC Driver for MySQL 5.x for Linux, page 414*
- *ODBC Driver for Aster Database for Linux, page 415*
- *ODBC Driver for DataDirect Cloud for Linux, page 417*
- *ODBC Driver for Amazon Redshift for Linux, page 418*
- *ODBC Driver for Vertica for Linux, page 420*
- *ODBC Driver for SAP HANA for Windows and Linux, page 421*
- *Other data sources and relational databases for Windows, page 424*

## MicroStrategy ODBC Driver for Apache Hive Wire Protocol for Windows and 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 Linux

The following information is required for setting up the driver connection for the MicroStrategy ODBC Driver for Amazon Redshift Wire Protocol:

- **Data Source Name:** A name to identify the Amazon Redshift data source configuration in MicroStrategy. For example, `Finance` or `Redshift-1` can serve to identify the connection.
- **Host Name:** The server name or IP address of the machine on which the Amazon Redshift data source resides. Contact your system administrator for the server name or IP address.
- **Port Number:** The port number for the connection. In most cases, the default port number is **5439**, but you should check with your database administrator for the correct number.
- **Database Name:** The name of the database to connect to by default. The database administrator assigns the database name.

## MicroStrategy ODBC Driver for DB2 Wire Protocol for Windows and Linux

The following information is required for setting up the driver connection for MicroStrategy ODBC Driver for DB2 when running against 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 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 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 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 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 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:

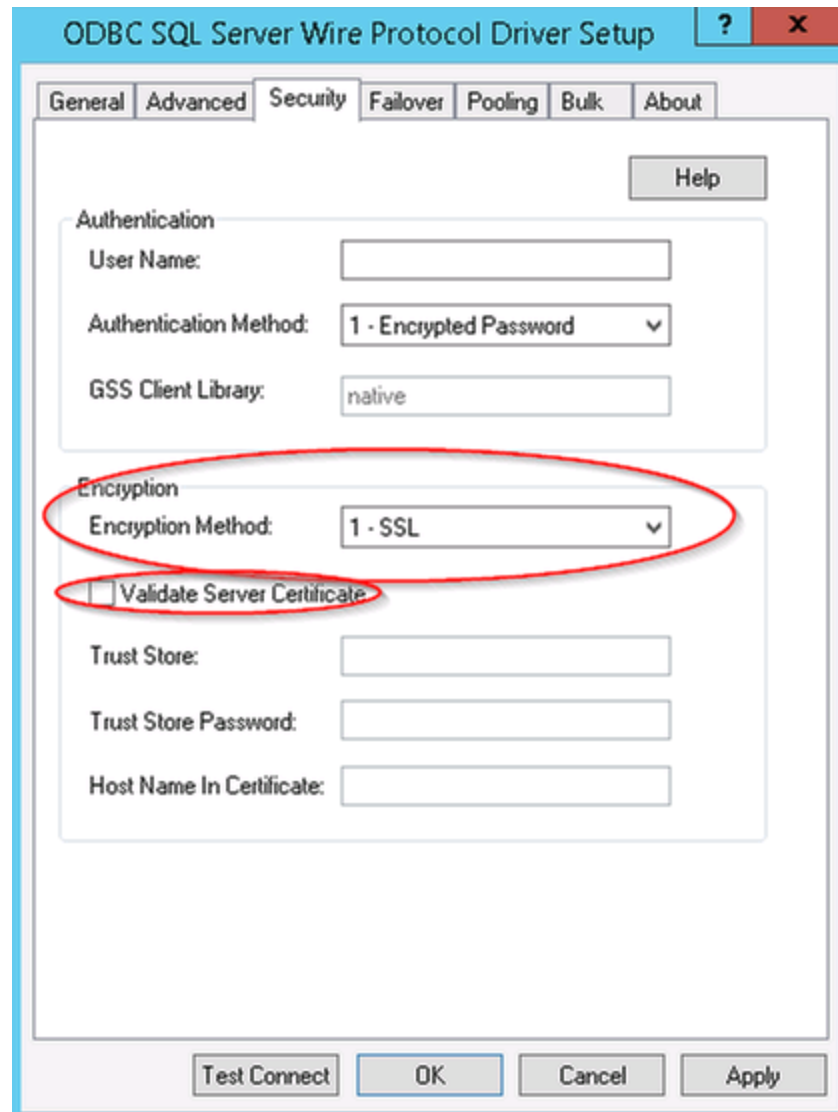
- `SQLServer-1\Instance1`
- `123.45.678.998\Instance1`
- **Port Number:** The port number for the connection. The default port number for SQL Server is usually **1433**. Check with your database administrator for the correct number.
- **Database Name:** The name of the database to connect to by default. 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.
- **Enable SSL encryption:**
  - **Windows:** Open the DSN in the ODBC Administrator and edit the Security tab so that **Encryption Method** is set to 'SSL' and the **Validate Server Certificate** is unchecked as shown below.



- **Linux:** Edit the `odbc.ini` file and set the following values in the DSN:
  - `EncryptionMethod=1`
  - `ValidateServerCertificate=0`

## MicroStrategy ODBC Driver for MongoDB for Windows and Linux

The following information is required for setting up the driver connection for the MicroStrategy ODBC Driver for MongoDB:

- **Data Source Name:** A name to identify the MongoDB data source configuration in MicroStrategy. For example, `Finance` or `MongoDB-1` can serve to identify the connection.



- **Host Name:** The server name or IP address of the machine on which the MongoDB data source resides. Contact your system administrator for the server name or IP address.
- **Port Number:** The port number for the connection. Check with your database administrator for the correct number.
- **Database Name:** The name of the database to connect to by default. The database administrator assigns the database name.
- **Schema Definition Path:** The path where configuration files that define the relational map of native data are stored. These configuration files are created in this location when first connecting to the data source and then used for subsequent connections.

## MicroStrategy ODBC Driver for Oracle Wire Protocol for Windows and Linux

The following information is required for setting up the driver connection for MicroStrategy ODBC driver for Oracle Wire Protocol:

**Data Source Name:** Enter a name to identify the Oracle data source configuration in MicroStrategy. For example, Finance or Oracle-1 can serve to identify the connection. A DSN is required for any Oracle Wire Protocol connection. Depending on whether you want to use a standard connection or a TNSNames connection, refer to one of the following lists of options below:


- **Standard Connection:** A standard connection is configured through Oracle Wire Protocol with the following connection parameters:
  - **Host Name:** The name of the Oracle server to be accessed. 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.

When connecting to an installation of Oracle as metadata, it is imperative that the ODBC driver is configured to use the character set parameters as the Oracle metadata installation. This is controlled through the IANAAppCodePage option in the odbc.ini file.

Follow these steps to determine the character set of your Oracle metadata installation and adjust the odbc.ini files:

-  1 Open the MicroStrategy ODBC Test Tool and connect to your database.
- 2 Run the following query `select * from NLS_DATABASE_PARAMETERS.`
- 3 Note the value returned for the NLS\_CHARACTERSET parameter.
- 4 Find the NLS\_CHARACTERSET value in the IANA Registry found at <http://www.iana.org/assignments/character-sets/character-sets.xhtml>.
- 5 Open the odbc.ini file and set the IANAAppCodePage value to the assigned MIBenum value from the IANA registry.

## MicroStrategy ODBC Driver for Pivotal Greenplum Wire Protocol for Windows and 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 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 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 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 404](#)
- [Preparing the Microsoft Excel file, page 405](#)
- [Configuring the MicroStrategy ODBC driver for SequeLink, page 406](#)

## Preparing the Microsoft Access database

You must complete the steps below to access an Access database stored on a Windows machine from an Intelligence Server hosted on a Linux machine.

### 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 161](#).

## To access Microsoft Access databases from an Intelligence Server hosted on 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 86](#)).



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 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 406](#).

## Preparing the Microsoft Excel file

You must complete the steps below to access Excel files stored on a Windows machine from an Intelligence server hosted on a 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 425](#).
- 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 426](#).

---

## To access Microsoft Excel files from an Intelligence Server hosted on 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 86](#)).

## 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 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 406](#).

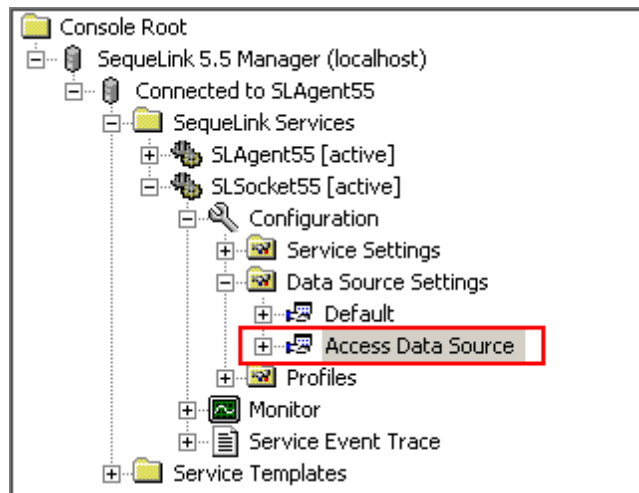
## Configuring the MicroStrategy ODBC driver for SequeLink

The steps below show you how to configure the MicroStrategy ODBC driver for SequeLink to access either Microsoft Access databases or Excel files stored on a Windows machine.

## To configure the MicroStrategy ODBC driver for SequeLink

- 1 On the Linux machine that hosts Intelligence Server, browse to `HOME_PATH` where `HOME_PATH` is the directory you specified as the Home Directory during installation.

- 2 Open the `odbc.ini.example` file and find the section that starts with `[SequeLinkODBC]`. Copy this section into the `odbc.ini` file. For information on the parameters, refer to DataDirect's documentation at <http://media.datadirect.com/download/docs/odbc/allodbc/help.html>.
- 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:



- 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 271](#).

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 Linux

The ODBC driver for Red Brick is not a MicroStrategy-branded driver. 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, see the *Installation and Configuration Guide for UNIX and Linux* provided by IBM.



Be sure to 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 ODBC Lib (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 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 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, see the product documentation provided directly by the database vendor.

## ODBC Driver for Sybase Adaptive Server IQ for 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 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 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:

```
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:

```
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 installation.
- 7 Open the `odbc.ini.example` file and find the section that starts with `[SAPSYBASEIQVersion]`, 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.
- 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 Teradata for Linux

ODBC driver for Teradata is not a MicroStrategy-branded driver. The following steps show how to configure the ODBC driver for Teradata.

For information on setting up an ODBC driver for Teradata through the Connectivity Wizard on Windows, see [Other data sources and relational databases for Windows, page 424](#).

### 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.



The directory where the ODBC driver for Teradata is installed should always be

accessible to MicroStrategy Intelligence Server.



You can enable Teradata Parallel Transporter for your connections to Teradata. This can improve performance when retrieving large amounts of data, typically 0.5 Gigabytes and larger, which can occur most commonly in MicroStrategy when publishing Intelligent Cubes. For steps to configure this support, refer to the [MicroStrategy Web Help](#).

### To configure a DSN

- 2 In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.
- 3 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.
- 4 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.

- 5 Save the `odbc.ini` file.

## ODBC Driver for Informix 8 for Linux

The MicroStrategy ODBC Driver for Informix 8 is already installed in the `INSTALL_PATH/lib` directory.



The ODBC Driver for Informix 8 for Linux is a MicroStrategy-branded ODBC driver, but it is not accessible through the Connectivity Wizard.

However, the Informix Client Software Developer's Kit (CSDK) must be installed before you create a DSN. This software is not included in the MicroStrategy product suite installation and must be obtained through the database vendor or a third party. For information on installation, refer to the product documentation provided directly by the database vendor.

The following steps show how to configure the MicroStrategy ODBC driver for Informix 8.

---

### To configure ODBC Driver for Informix 8

---

- 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 Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified 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 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:

```
chmod a-w ODBC.sh
```

## To configure a DSN

- 6 In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified 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 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 Linux console window, browse to `HOME_PATH`, where `HOME_PATH` is the directory you specified as the home directory during installation.
- 3 Edit the `odbcinst.ini` file and replace all instances of `<NETEZZA_ODBC_DIR>` with the location of the directory where the Netezza ODBC Driver is installed. An example of this is as follows:

If the original path is:

```
Driver = /<NETEZZA_ODBC_DIR>/lib64/libnzodbc.so
```

Then the modified path will be:

```
Driver = /usr/odbc/netezzahome/lib64/libnzodbc.so
```

- 4 Save the `odbcinst.ini` file.

### To modify the `odbc.ini` file

- 5 Open the `odbc.ini.example` file and search for the section that starts with `[IBM Netezza]`.
- 6 Open the MicroStrategy `odbc.ini` file.
- 7 Copy and paste the contents from the `odbc.ini.example` file for your Netezza ODBC driver. You should paste the contents of the DSN exactly as they appear in the example file.
- 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 `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 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='<MYSQL_PATH>'
```

Replace this `<MYSQL_PATH>` placeholder with the location of where the MySQL driver is installed. Do not modify any other occurrences of `<MYSQL_PATH>` within `odbc.sh`.

- 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 specified 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 203](#) and [Creating a database connection, page 205](#).

## 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 specified 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 specified 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.



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 specified 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 specified 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 Linux, page 396](#)). 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 specified 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 make the following changes:

- Provide the location of the directory where the Amazon Redshift ODBC driver files are installed. The following definition is included:

```
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.redshiftodbc.ini`. The following definition is included:

```
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:

```
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 specified 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

The ODBC driver for Vertica is not a MicroStrategy-branded driver. The following steps show how to configure the ODBC driver for Vertica for Linux.

You must modify the `odbc.ini` file to create the DSN for Vertica.



The third-party product(s) discussed in the procedure below is manufactured by vendors independent of MicroStrategy. MicroStrategy makes no warranty, express, implied or otherwise, regarding this product, including its performance or reliability.

### To configure ODBC driver for Vertica

- 1 Install the ODBC Driver for Vertica for the Linux operating system. For information on installation, refer to the product documentation provided by the database vendor.



- The path to the installation location you choose for the ODBC driver is used later in this procedure as the value for the Driver parameter in the `odbc.ini` file.
- 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 specified 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 `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 console window, browse to `HOME_PATH` where `HOME_PATH` is the specified 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 422](#)
- [To configure an ODBC Driver for SAP HANA on Linux, page 422](#)

### 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 170](#) 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 163](#).


- 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 362](#).
- 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 203](#).

 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 424](#).

## 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 specified 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 where you installed the SAP HANA ODBC driver files. Within the `ODBC.sh` file, the following definition is included:

```
SAPHANA_PATH='<SAPHANA_PATH>'
```

Replace this `<SAPHANA_PATH>` placeholder with the location of the SAP HANA ODBC driver files. Do not modify any other occurrences of `<SAPHANA_PATH>` within `odbc.sh`.

- 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 Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during 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 `port` with the port number for the connection to the SAP HANA database. Contact your SAP HANA database administrator for the required port number.

- `USER=uid`

Replace `uid` with a valid SAP HANA user account.

- `PASSWORD=pwd`

Replace `pwd` with the password for the SAP HANA user account described above.

For information on the available parameters, refer to your third-party SAP HANA driver documentation. This can often be found along with the driver installation.

- 9 Save the `odbc.ini` file.
- 10 You can use the MicroStrategy DB Query Tool to test whether data can be retrieved from your SAP HANA data source. For information on how to use the MicroStrategy DB Query Tool, see [Using the DB Query Tool, page 362](#).

- 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 [Creating a database instance, page 203](#).



When creating a database connection, which is part of a database instance, for SAP HANA, ensure that you select Non UTF-8 as the character set encoding for Linux drivers.

This completes the steps for the initial connection to SAP HANA in MicroStrategy for Windows environments. For additional configuration requirements, see [Additional requirements to support SAP HANA, page 424](#) below.

## Additional requirements to support SAP HANA

Review the following additional requirements to ensure a successful integration of SAP HANA in MicroStrategy:

- Be aware that once you import tables from SAP HANA into MicroStrategy, you must manually import any prefix information as well. Using the Warehouse Catalog, you can select all tables imported from SAP HANA and select Import Prefix to import the prefix information. For additional steps to access and use the Warehouse Catalog, see the [Project Design Guide](#).
- If the tables in SAP HANA include input parameters, these are supported in MicroStrategy using prompts. Using the Table Editor available in MicroStrategy Developer, you can create and modify prompts to support input parameters. For steps to access and use the Table Editor, refer to the *Project Design Help*.

## Other data sources and relational databases for Windows

If you use other databases or data sources, refer to the database-specific documentation for information on required settings. Standard settings are supported by MicroStrategy for most relational databases.

### 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 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.



The Name Box is the drop-down list on the left-hand side below the toolbars.

You can create multiple tables in one Excel file by highlighting different parts of the file and assigning them different names.

- 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 203](#).

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 392](#).
    - 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 392](#).
      - 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 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 72](#).

These ODBC DSNs are defined by specifying values for certain DSN parameters. This file is activated by the environment variable `ODBCINI`, and is required by all ODBC applications. By default, the `odbc.ini` file is installed in `HOME_PATH`, where `HOME_PATH` is the directory you specified as the home directory during installation on Linux. It contains the definitions for all 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 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 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
- The following communication ports are open:
  - 22 - Hadoop Gateway Host
  - 88 - Kerberos
  - 1006 - publish txt files
  - 4020 - Hadoop Gateway Status
  - 8020 - Namenode
  - 10000 - Hive
  - 10006 - Cluster worker node
  - 10015 - SparkSQL
  - 30004 - Hadoop Gateway Host (inbound)
  - 30241 - MicroStrategy Server (outbound)
  - 50070 - WebHDFS

## 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](#)

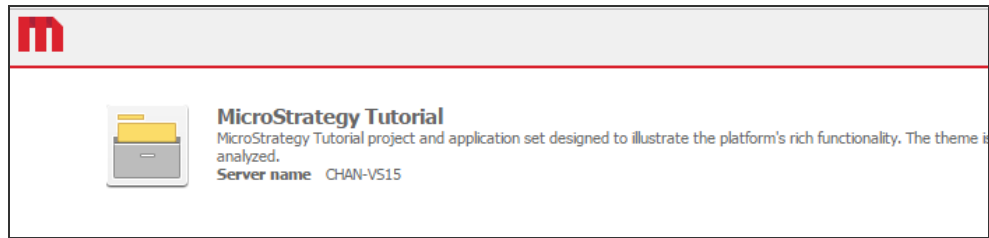
---

### Deploy MicroStrategy Hadoop Gateway via Gateway Manager

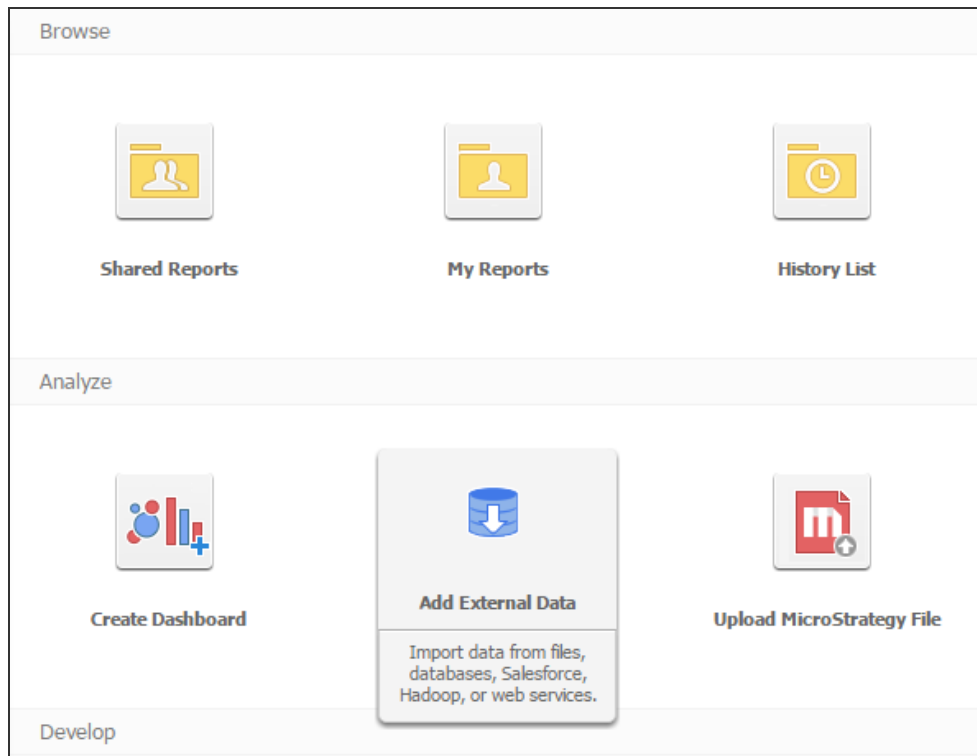
---

- 1** Login to MicroStrategy Web with administrator privilege and open a specific project.

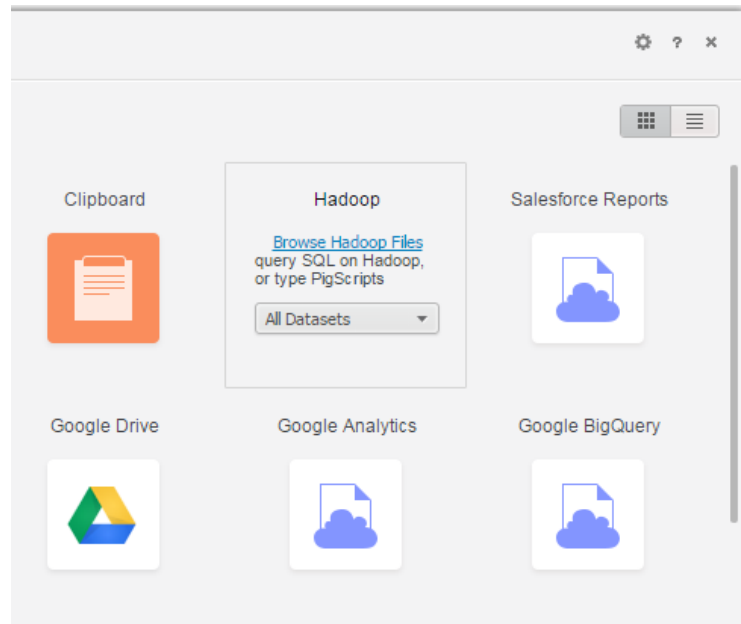




**2 Click Add External Data.**



**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](#).

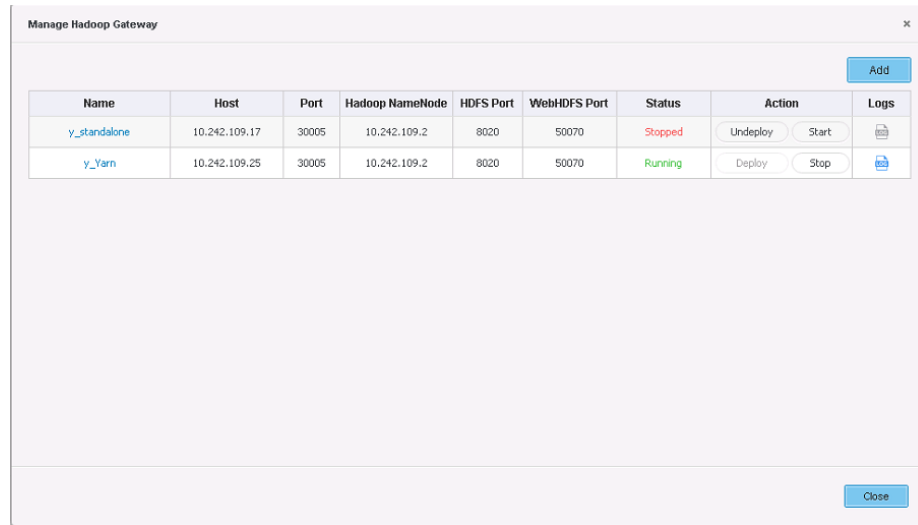
 A screenshot of the 'Data Source' dialog box. The 'Database' dropdown is set to 'Hadoop', 'Version' to 'Big Data Engine', and 'Big Data Engine' to 'spark\_local'. A red arrow points to a blue link labeled 'Manage Hadoop Gateway' next to the 'Big Data Engine' dropdown. Below this, there are text input fields for 'Hadoop Name Node' (10.242.109.2), 'HDFS Port' (8020), 'WebHDFS Port' (50070), 'Big Data Engine' (10.197.36.79), 'Port' (30004), 'Authentication mode' (Anonymous), 'User' (user), 'Password' (masked with dots), and 'Data Source Name' (Hadoop). At the bottom right, there are 'OK' and 'Cancel' buttons. A 'Show connection string' link is also visible.

To configure file retrieval using HTTPS click **Show connection string** and add 'Protocol=HTTPS' as shown below

The screenshot shows a 'Data Source' configuration window. The 'Database' is set to 'Hadoop' and the 'Version' is 'Big Data Engine'. The 'Hadoop Gateway' dropdown is empty, with a 'Manage Hadoop Gateway' link next to it. The 'Hadoop Name Node' is '10.242.109.2'. The 'Protocol' is 'HTTP'. The 'HDFS Port' is '8020', 'WebHDFS Port' is '50070', 'Host' is '10.20.127.19', 'Port' is '12062', and 'Authentication mode' is 'Anonymous'. The 'Edit connection string' checkbox is checked. The connection string text area contains: `hadoopName=10.242.109.2; hdfsPort=8020; webHDFSPort=50070; BDEIP=10.20.127.19; BDEPORT=12062; AUTHMODE=Anonymous; Protocol=HTTPS;`. The 'Data Source Name' is 'Penguin cluster'. There are 'OK' and 'Cancel' buttons at the bottom right.

Database:	Hadoop
Version:	Big Data Engine
Hadoop Gateway:	<a href="#">Manage Hadoop Gateway</a>
Hadoop Name Node:	10.242.109.2
Protocol:	HTTP
HDFS Port:	8020
WebHDFS Port:	50070
Host:	10.20.127.19
Port:	12062
Authentication mode:	Anonymous
<input checked="" type="checkbox"/> Edit connection string	<a href="#">Hide connection string</a>
<code>hadoopName=10.242.109.2; hdfsPort=8020; webHDFSPort=50070; BDEIP=10.20.127.19; BDEPORT=12062; AUTHMODE=Anonymous; Protocol=HTTPS;</code>	
Data Source Name:	Penguin cluster

The Gateway Manager appears and lists all managed Hadoop Gateways.



- 6 To add a Hadoop Gateway, click **Add**. The Hadoop Gateway dialog box appears. If you enabled HTTPS file transfer in the previous step, select the Use HTTPS check box.

**Hadoop Gateway** [X]

Name: David cluster

*Hadoop Properties*

Hadoop NameNode: 10.242.109.2

☐ Use HTTP5

HDFS Port: 8020

WebHDFS Port: 50070

*Gateway Properties*

Host: 10.242.109.10

Port: 30004

Path: /opt

*Spark Properties*

Mode: YARN

Authentication mode: Anonymous

JAR of path: hdfs://10.242.109.2:8020/jar-path-cloudera/spark-assembly-

[Advanced](#) ▾

[Delete] [Save] [Cancel]

- a 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 Next to Hadoop Gateway, click **Download Now**. The HGoS Manager package is download.

- 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=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

```

## 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.

# How to manually deploy Hadoop Gateway if the Hadoop cluster has Kerberos authentication enabled

After Hadoop Gateway is installed and before starting Hadoop Gateway Manager:

1. Log via console to the Hadoop Gateway host and create a directory `HGOS_HOME/krb5/` to store Kerberos keytab files. Ensure the Hadoop Gateway user `hgos` has write privileges in this new folder.
2. Generate a new valid keytab file by following these commands



```
mkdir HGOS_HOME/krb5
ktutil
add_entry -password -p
hgos/<HadoopGatewayHostFQDN>@REALM_NAME -k 1 -e
aes256-cts-hmac-sha1-96
wkt HGOS_HOME/krb5/hgos.keytab
exit
```

3. After replacing the keytab file and principal name, create the key cache file by executing the line below:

```
mkdir HGOS_HOME/log && chmod -R 777 HGOS_HOME/log
kinit -k -t <your keytab filepath> <your principal
name> -c log/krb5cc_hdfs
```

4. Add the following lines to the file `/opt/hgos/conf/hgos-spark.properties`:

```
spark.yarn.token.renewal.interval=50000
spark.yarn.security.tokens.hdfs.enabled=true
spark.yarn.principal=
hgos/<HadoopGatewayHostFQDN>@REALM_NAME
spark.yarn.keytab=HGOS_HOME/krb5/hgos.keytab
```

5. Save your changes and close the file.



To confirm the Intelligence Server has a valid token to browse WebHDFS, execute the following command to force reading Kerberos tokens in the user session:

```
curl --negotiate -u :
https://YourNameNode:50470/webhdfs/v1/?OP=liststatus
```

## Import Data from Hadoop

For instructions to import data from Hadoop, see *Importing data from Hadoop* in the [MicroStrategy Web Help](#).



The connectivity timeout between Hadoop Gateway and Intelligence Server is 20 minutes by default. To increase this timeout limit, create a file named `QueryDSServerTimeout.ini` and place it in your Intelligence Server directory. The only entry in this file will be the numeric value (in minutes) for your timeout limit. Placing a value of -1 in this file will set the timeout to unlimited.

# Troubleshooting

This appendix provides information on common problems that you might encounter while installing and configuring MicroStrategy on Linux and Windows operating systems.

## Reviewing general installation errors

Any errors in your MicroStrategy installation are written to the `install.log` file.

---

### Review errors found in `install.log`

---

- 1 Browse to `INSTALL_PATH` where `INSTALL_PATH` is the directory you specified as the Install Directory during installation
- 2 Open the `install.log` file. (For Unix, use an editor with a command like `dtpad install.log`. For Windows, use Notepad or some other text editor to open the file.)
- 3 Review the error messages. A common error is to run out of space.

## Graph and document support of non-Western European fonts

If your Linux system uses non-Western European fonts, you may see indiscernible values returned in place of text on your graphs when accessed through MicroStrategy Web. This also occurs for Report Services documents when accessed through MicroStrategy Web or any other MicroStrategy client application (such as Developer) that connects to an Intelligence Server connected (three-tier) project source.

To support non-Western European fonts, copy True Type fonts into the Intelligence Server installation directory. Copy these fonts, which have a `.ttc` or `.ttf` extension, to `INTELLIGENCE_SERVER_INSTALL_PATH\PDFGeneratorFiles`. The default installation path for the Intelligence Server in Linux is

home\MicroStrategy\PDFGeneratorFiles. For the change to take effect, you must restart Intelligence Server.

## Server port number errors

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.
  - **Linux:** In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation. Browse to the folder `bin` and type `./mstrsvcmgr`, then 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 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 170.

### Creating a metadata fails due to insufficient page size

When creating a metadata within a IBM DB2 database, the following error and resulting error message can be encountered:

```
Invalid Page Size for DB2 UDB Metadata
```

This can occur when an IBM DB2 database user has access to a tablespace that does not have the space requirements necessary for the metadata repository. The MicroStrategy metadata repository requires a page size of at least 8 KB.

To avoid this error, it is recommended to revoke access to any tablespaces that are lower than 8 KB in page size for the user account that is creating the metadata. The user must also have access to a tablespace with a page size of at least 8 KB. Contact your database administrator to perform this user configuration.

### Creating a History List repository fails due to insufficient page size

When creating a History List repository within a Sybase database, the following error and resulting error message can be encountered.

```
Creating table TableName failed because the minimum row
size would be PageSizeRequirement bytes. This exceeds
the maximum allowable size of a row for this table,
PageSizeLimit bytes.
```

This can occur when a Sybase database does not have the page size requirements necessary for the History List Repository. To solve this error, contact your database administrator to increase the current available space (*PageSizeLimit* in the error message above) for the table space to be large enough to store the History List repository (*PageSizeRequirement* in the error message above). As a general rule, the page size for the History List repository should be at least 4 KB.

## Permission errors

This section provides troubleshooting information on permission errors in a Linux environment.

### Missing JVM file

The installation fails just before it starts transferring files, and the following error is displayed:

JVM not found

Clear the set group ID on execution(s) bit on the permissions of the directory where the InstallPath for the Intelligence Server is to be placed.

- 1 In a console window, type:

```
#chmod g-s directory
```

where *directory* is the InstallPath for Intelligence Server.

- 2 Press ENTER.

# USHER ADMINISTRATION

This section describes processes that enable administrators to manage Usher.

- [Managing Usher Administrators](#)
- [Managing the Usher Signing Certificate Authority](#)

## Managing Usher Administrators

After Usher is installed and configured, you provisioned the first Usher Server administrator badge (referred to as *Usher Admin*). All Usher Admins have the ability to log into Usher Network Manager to access the Usher Configuration, create new Usher Networks, and manage other Usher Admins.

---

### To add an Usher Admin

---

- 1 Log in to Usher Network Manager:
  - a In a web browser, navigate to your organization's Network Manager home page. The Usher Network Manager page opens.
  - b On your smartphone, open the Usher Security app, then open the QR code reader.
  - c Scan the QR code displayed on the Usher Network Manager page to log in to your network.
- 2 Click the **Administrator** button in the upper-right. The Administrators page opens.
- 3 Click the + **Add Administrator(s)** button in the upper-right.
- 4 In the pop-up, enter the email address of the new Usher administrator and click **Add**.
- 5 In the next pop-up, enter the first name and last name of the new Usher administrator and click **Add**. The new Usher administrator is added to the list of administrators. Additionally, the new administrator is notified and provided instructions for getting the Usher Admin badge.

---

## To delete an Usher Admin

---


- 1 Log in to Usher Network Manager:
  - a In a web browser, navigate to your organization's Network Manager home page. The Usher Network Manager page opens.
  - b On your smartphone, open the Usher Security app, then open the QR code reader.
  - c Scan the QR code displayed on the Usher Network Manager page to log in to your network.
- 2 Click the **Administrator** button in the upper-right. The Administrators page opens.
- 3 Click the check box next to the Usher Admin(s) that you want to delete.
- 4 Click the - **Delete Administrator(s)** in the upper-right.
- 5 In the pop-up, click **Delete**.

## Managing the Usher Signing Certificate Authority

The Usher Server uses a self-signed SSL certificate to function as a Certificate Authority (CA) for the purpose of signing client certificates used by other Usher components, such as the Usher Agent and custom applications implementing Usher APIs.

- For **Windows** implementations, the MicroStrategy Installation Wizard automatically generates an Usher Signing CA certificate valid for 1 year, and propagates it to the required system locations so that Tomcat will trust this CA.
- For **Linux** implementations, you must generate and propagate the Usher Signing CA certificate manually.
- Regardless of the host, when the Usher Signing CA certificate expires or is about to expire, you must manually regenerate and propagate a new certificate.

You can create the Usher Signing CA using OpenSSL®.

 If you copy and paste the OpenSSL commands shown below, please ensure that the option flags are marked with a hyphen/minus sign instead of an en dash or em dash. If any hyphens are autoformatted into different characters, then they will be parsed incorrectly on the command line and you will get "unrecognized option" errors.

- On **Windows**, an OpenSSL utility is installed along with Usher.

The default file locations are as follows:

- Usher Signing CA certificate: `C:\Program Files (x86)\Common Files\MicroStrategy\certificates\ushersigningca.crt`

- Usher Signing CA private key: `C:\Program Files (x86)\Common Files\MicroStrategy\keys\ushersigningca.key`
- 64-bit Java CA store: `C:\Program Files (x86)\Common Files\MicroStrategy\JRE\180_77\Win64\lib\security\cacerts`
- OpenSSL executable: `C:\Program Files (x86)\Common Files\MicroStrategy\OpenSSL\openssl-1.0.2e\openssl.exe`
- On **Linux**, an openssl utility is included with many distributions.

---

## Generate a new Usher Signing Certificate Authority

---

Follow these steps if you do not already have an Usher Signing CA private key and certificate (for example, if you are setting up Usher for the first time on Linux).

### 1 Use OpenSSL to generate a private key (.key) and certificate (.crt).

- **Windows:**

- a Run Command Prompt as administrator.
- b Navigate to the folder where the OpenSSL executable is located.

```
> cd C:\Program Files (x86)\Common
Files\MicroStrategy\OpenSSL\openssl-1.0.2e
```

- c Run the OpenSSL executable.

```
> openssl.exe
```

- d Create the request for a new private key and certificate. Following is a sample command that creates a 3072-bit key and a certificate valid for 365 days. If you want a larger key size or a longer certificate lifetime, update the command appropriately.

```
> req -newkey rsa:3072 -nodes -new -x509 -days 365 -
subj '/C=US/O=Usher Server/CN=Usher Server ROOT Signing
CA' -keyout <PathToCertFolder>/ushersigningca.key -out
<PathToCertFolder>/ushersigningca.crt
```

- **Linux:**

- a Log into the command line.
- b Run the OpenSSL utility to create the request for a new private key and certificate. Following is a sample command that creates a 3072-bit key and a certificate valid for 365 days. If you want a larger key size or a longer certificate lifetime, update the command appropriately.

```
> openssl req -newkey rsa:3072 -nodes -new -x509 -days
365 -subj '/C=US/O=Usher Server/CN=Usher Server ROOT
Signing CA' -keyout
<PathToCertFolder>/ushersigningca.key -out
<PathToCertFolder>/ushersigningca.crt
```



---

## Regenerate a new Usher Signing Certificate Authority Certificate

---

Follow these steps if you have an Usher Signing CA private key and need to generate a new certificate (for example, your existing CA certificate has expired or is about to expire).

- 1 Use OpenSSL to generate a new certificate (.crt) based on your existing private key (.key).
  - **Windows:**
    - a Open a command prompt window as administrator.
    - b Navigate to the folder where the OpenSSL executable is located.

```
> cd C:\Program Files (x86)\Common
Files\MicroStrategy\OpenSSL\openssl-1.0.2e
```
    - c Run the OpenSSL executable.

```
> openssl.exe
```
    - d Create the request for a new certificate. Following is a sample command that generates a certificate valid for 365 days. If you want a longer certificate lifetime, update the command appropriately.

```
> req -new -x509 -days 365 -subj '/C=US/O=Usher
Server/CN=Usher Server ROOT Signing CA' -key
<PathToCertFolder>/ushersigningca.key -out
<PathToCertFolder>/ushersigningca.crt
```
  - **Linux:**
    - a Open a terminal window.
    - b Run the OpenSSL utility to create the request for a new certificate. Following is a sample command that generates a certificate valid for 365 days. If you want a longer certificate lifetime, update the command appropriately.

```
> openssl req -new -x509 -days 365 -subj '/C=US/O=Usher
Server/CN=Usher Server ROOT Signing CA' -key
<PathToCertFolder>/ushersigningca.key -out
<PathToCertFolder>/ushersigningca.crt
```

---

## Propagate the Usher Signing Certificate Authority

---

- 1 Append the Usher Signing CA certificate to the Certificate Authority Chain (.pem) file used by your Usher environment.
  - **Windows:**
    - a Run the following command:

```
> type "<PathToCertFolder>/ushersigningca.crt" >>
"<PathToCertFolder>/UsherCAChain.pem"
```

- **Linux:**

- a Run the following command:

```
> cat <PathToCertFolder>/ushersigningca.crt >>
<PathToCertFolder>/UsherCAChain.pem
```

- 2 To ensure that Tomcat will trust the Usher Signing CA, you must add it to the CA store of the Java Runtime Environment (JRE) used by Tomcat.

- **Windows:**

- a Run Command Prompt as administrator.

- b Run the following command:

```
> "C:\Program Files (x86)\Common
Files\MicroStrategy\JRE\180_77\Win64\bin\keytool.exe" -
import -alias <NewAlias> -file
"<PathToCertFolder>\ushersigningca.crt" -keystore
"C:\Program Files (x86)\Common
Files\MicroStrategy\JRE\180_
77\Win64\lib\security\cacerts"
```

- c Enter the passphrase when prompted (the default is "changeit").
  - d Confirm you want to add the certificate to the store when prompted.

- **Linux:**

- a Log into the command line.

- b Run the following command:

```
> keytool -import -alias <NewAlias> -file
<PathToCertFolder>/ushersigningca.crt -keystore
<PathToJavaRoot>/lib/security/cacerts
```

- c Enter the passphrase when prompted (the default is "changeit").
  - d Confirm you want to add the certificate to the store when prompted.

---

## Update the Usher Configuration

---

- 1 If you previously completed the Usher post-install configuration (see the section [To complete the Usher Configuration \(Windows and Linux\)](#)), and the new Usher Signing CA certificate or private key do not have the same file name or path as the old ones, you must update those fields in Usher Configuration.
  - a In a web browser, navigate to Usher Network Manager.
  - b Use your Usher Admin badge to scan the QR code and login.
  - c From the drop-down menu in the upper-right corner, click **Usher Configuration**.

- d Update the **SSL Certificate Authority Certificate** and **SSL Certificate Authority Key** fields to the new paths.
  - e Click **Next** to apply the changes.
- 2 If the Usher Security Server or Usher Gateway Server web services are running, restart them.
  - 3 If you have any apps using Usher client certificates, you must regenerate those client certificates so they are signed by the new Usher Signing CA and replace them in the apps.