# CONTENTS

**Planning Your Upgrade** ................................................................. 4

- Determine Your Upgrade Methodology ......................................... 4
- Pre-Upgrade Administrative Tasks ................................................ 5
- Upgrade Prerequisites .................................................................. 7
- Upgrade Best Practices .................................................................. 8
- The Upgrade Process Checklist ....................................................... 10

**Executing the Upgrade** ................................................................ 12

- Performing an In-Place Upgrade .................................................. 13
- Performing a Parallel Upgrade ..................................................... 68
- Using MicroStrategy Backup and Restore ....................................... 120

**Validating the Upgrade** ............................................................... 155

- URL Scanner - Identify URLs for Export ...................................... 155
- Integrity Manager - Upgrade Impact Testing .................................. 159
- MicroStrategy Capacity Testing Tool ............................................ 167
- Upgrade Analysis Dossier ............................................................. 200
- Resolving Problems Encountered During the Upgrade ..................... 211

**Supplemental Information** ............................................................ 213

- Application Servers ..................................................................... 213
- Using a Response File with Configuration Wizard ......................... 288
- Defining DSNs ............................................................................. 312
- Specifying Warehouse Connection Information ............................ 316
- Minimizing the Impact of Data Changes for Platform Release Upgrades ......................................................... 335
Planning Your Upgrade

Upgrading to MicroStrategy 2020 can have a significant effect on your system. New features and enhancements may cause changes to user workflows as well as the data returned by report executions. To ensure a successful upgrade process MicroStrategy strongly recommends that you carefully review the upgrade information covered in this section before beginning your upgrade.

Determine Your Upgrade Methodology

Before you begin the upgrade process, you will need to decide on an upgrade methodology to follow. Choose the best option based on the size, complexity, and acceptable downtime for your particular environment.

- **In-Place Upgrade**: The upgrade is made directly to the current in-place environment hardware. Your MicroStrategy environment will be unavailable to end users throughout the upgrade process. This option works best for smaller deployments and those where downtime is acceptable.

  You do not need to uninstall your current version of MicroStrategy before upgrading. MicroStrategy recommends performing an over-install when doing an in-place upgrade.

- **Parallel Upgrade**: The upgrade is first executed on a test environment. The testing environment is configured to mirror the current production environment. Once the upgrade process, data validation, and performance tests are completed and evaluated the testing environment can be configured for use as the new production environment.
Pre-Upgrade Administrative Tasks

Before you begin upgrading any MicroStrategy systems ensure the following actions have been taken.

System Preparation

- All necessary hardware (if applicable) for your new environment has been procured.
- All necessary information from all hardware has been collected and is available, including:
  - Fully Qualified Domain Names and IP Addresses
  - Machine credentials
  - Network Firewalls
  - Installation paths that will be used
  - Server type (.NET or J2EE)
- Review the Readme for your new version of MicroStrategy to ensure that all minimum system requirements are met.
- If you are planning to upgrade Enterprise Manager, use Command Manager to run an Enterprise Manager data load before you upgrade Intelligence Server or you may lose access to some statistics data. Make sure the Close orphan sessions option is selected.

Backup Repositories and Stores

Upgrading to a new feature release or platform release of MicroStrategy will require the metadata to be updated, and additionally may require updating other repositories and stores including:

- Identity Store
- Collaboration Store
MicroStrategy recommends that administrators back up metadatas, stores, and repositories before they begin the actual production migration for fallback support.

Backup Configuration Files

MicroStrategy Web

Backup any changed MicroStrategy Web configuration files:

- ../WEB-INF/xml/config/mapConfig.xml
- ../WEB-INF/xml/sys_defaults.properties
- ../WEB-INF/xml/sys_defaults.xml
- ../WEB-INF/web.xml

MicroStrategy Mobile

Backup any changed MicroStrategy Mobile configuration files:

- /WEB-INF/xml/sys_defaults.properties
- /WEB-INF/xml/sys_defaults.xml
- /WEB-INF/web.xml

Backup any mobile device configurations or saved any images from the MicroStrategy Photo Uploader widget, manually back them up before proceeding with your upgrade.

- /WEB-INF/xml/mobile
- /WEB-INF/SavedImages
If you secure your MicroStrategy Mobile connections through Certificate Server, you must upgrade your Certificate Server to support MicroStrategy 2020 Mobile applications.

MicroStrategy Library

Backup any MicroStrategy Library Configuration Files:

- /webapps/MicroStrategyLibrary/images/
- /webapps/MicroStrategyLibrary/plugins/
- /webapps/MicroStrategyLibrary/WEB-INF/classes/auth/
- /webapps/MicroStrategyLibrary/WEB-INF/classes/config/configOverride.properties
- Any other customizations including SSL configurations

MicroStrategy Collaboration

Back up the Collaboration Server notification files in the following directory:

<INSTALL_PATH>\MicroStrategy\Collaboration Server\node_modules\mstr-collab-svc\pluginConfig\dossier

List of files:

- config.json
- email-Invite.template
- email-Mention.template

Upgrade Prerequisites

Make sure all of the following conditions are met before you continue:
Upgrade Guide

- Installation files have been downloaded from the MicroStrategy download site.
- Acquire a license key from MicroStrategy for the version of the MicroStrategy software that you are installing.
- New Windows machines will require a login account with administrative privileges for the domain or target machine
- Installing MicroStrategy on Linux with a CPU-based license key will require root access for installation

Upgrade Best Practices

Review the following recommendations to help ensure the success and stability of your MicroStrategy system and projects when upgrading to the latest version of MicroStrategy.

- The following versions of MicroStrategy are supported for upgrade directly to MicroStrategy 2020:
  - 10.4.x
  - 2019 (and related updates)

If you are currently on a version of MicroStrategy that is supported to directly upgrade to MicroStrategy 2020, you must first upgrade to a supported version before upgrading to MicroStrategy 2020.

- If you are currently on a version of MicroStrategy that is not supported to directly upgrade to 2020 and are doing an in-place upgrade, do not uninstall the current version of MicroStrategy before upgrading. MicroStrategy recommends performing an over-install.

- Review the Readme for a complete list of new products, new features, and updates in MicroStrategy.
Follow the upgrade order and recommendations outlined in this guide, in particular the *The Upgrade Process Checklist, page 10*. Always upgrade the Intelligence Server before upgrading client applications such as MicroStrategy Web, Mobile, Library, and Developer.

Create an upgrade test environment by duplicating your production environment and production metadata. Upgrade this test environment to validate the upgrade before using in your production environment. For guidelines on how to test your system, see *Validating the Upgrade*.

Never downgrade a machine from a newer version to an older version of MicroStrategy. It is not supported and can cause problems. Contact Tech Support if this has been accidentally done.

Avoid installing MicroStrategy products using virtual sessions on the host machine such as Windows Terminal Services. Always install MicroStrategy directly on the server machine's physical interface, or by using a remote connection tool (such as Microsoft Remote Desktop) that takes full control of the server machine's interface and creates an actual administrator session on the machine.

Shut down all MicroStrategy Intelligence Server nodes in a clustered environment before upgrading. For more information about clustering Intelligence Servers, see the *Clustering* chapter in the System Administration Guide.

Ensure that all data types assigned in existing projects are supported in the current MicroStrategy version. If a project containing columns with unsupported data types is upgraded, the data types for those columns are assigned as "reserved," and proper data types are not assigned in temporary tables. This affects report execution. Please see the Project Design Guide for a listing of the supported data types for each database type and additional information about changing to supported data types.
The Upgrade Process Checklist

The upgrade process described in the rest of this guide involves the following high-level steps. To help ensure a successful upgrade, follow these steps in the order they are presented in this guide.

1. Prepare the MicroStrategy system and projects for upgrade.
   - Review information specific to your version upgrade.
   - Pre-upgrade information and prerequisites.
   - Check for certified and supported configurations for all products and components being installed.
   - Back up the production metadata, repositories, and stores.
   - Create an upgrade test environment which is a duplicate of production.

2. Back up your customizations, if applicable.

3. If you are upgrading your Express installation environment, you can elect to keep the current version of tutorial or your existing database. During the upgrade, you are asked "Do you want to overwrite the existing database?"
   - Select Yes to overwrite the database and existing data.
   - Select No to keep your current database to use with the upgrade.


5. Using the test environment, update a copy of your product metadata.

6. Validate the test environment with functional testing, comparisons tests in Integrity Manager, and simulated end-to-end performance tests using the Capacity Testing Tool.

For more information, see Validating the Upgrade.
7. Upgrade and configure Intelligence Server in the production environment.

Executing the Upgrade

MicroStrategy recommends that you follow a specific product upgrade sequence when upgrading MicroStrategy products across several machines. Following this upgrade sequence helps ensure that MicroStrategy products installed on separate machines are interoperable, that is, they are able to work together when installed on separate machines.

The recommended upgrade sequence is as follows:

1. Intelligence Server


4. Remaining MicroStrategy Components:
   - **Client applications**
     - Desktop
     - Mobile and Library Mobile clients
     - Developer
     - Office
   - **Administrative applications**
     - Workstation
     - Command Manager
     - Enterprise Manager
     - Object Manager
     - Integrity Manager
Performing an In-Place Upgrade

Performing an In-Place Upgrade on a Windows Deployment

This section covers the procedure for directly upgrading your currently deployed Windows environment to MicroStrategy 2020. Carefully review the *Planning Your Upgrade* section before proceeding with your upgrade.

1. Install MicroStrategy 2020

Run the *MicroStrategy.exe* file included in your MicroStrategy 2020 download.

Step through each of the Installation Wizard windows:

1. Accept the MicroStrategy License Agreement.
2. Update your license key.
3. Review the list of products included in your license.
4. Review the products that will be installed.
   - If your license includes new products or services select them for installation in this window.
5. The installer will display any running services that need to be stopped. Click *Yes* to stop them now.
6. Configure the Platform Analytics connection to MySQL
8. Server activation information
9. Enter your contact information for server activation.

10. Choose to have an activation code sent to you now or at a later date.

11. Review the list of components you have selected for installation.

Choose if you would like the installer to automatically reboot the machine when needed.

12. Click **Install** to begin.

2. Update the Metadata

After upgrading the Intelligence Server, update the metadata by performing the procedure that follows. This procedure creates and updates metadata tables to support new features available in MicroStrategy 2020.

Updating the metadata ensure compatibility between your pre-existing objects and the MicroStrategy clients. You can also take advantage of new configuration settings exposed through the client applications, functionality, and objects available only with the updated version of metadata.

During the metadata update process, the Configuration Wizard generates additional SQL for tasks such as managing indexes and primary keys. To view this SQL during the update process, on the Summary Page of the Configuration Wizard, in the Summary pane, click **SQL Preview**. In addition, once you have completed the update process, the generated SQL is saved in the MicroStrategy common files directory. The file name is `OriginalScript_generated.sql`, where `OriginalScript` is the name of the original SQL update script. For example, if you are updating an Oracle database, the original SQL update script is `mdorcl.sql` and the generated SQL file is `mdorcl_generated.sql`.

Be aware of the following:

- If you have made any changes to the privileges assigned to the out-of-the-box MicroStrategy user groups and security roles, updating the metadata
may overwrite those changes. In particular, if you have made any changes to groups that use any privileges mentioned in the List of updated privileges, updating the metadata will overwrite those changes.

- For information about viewing the Configuration Wizard log file if an error occurs, see Resolving Problems Encountered During the Upgrade.

- The MDUpdate command line utility that was formerly used to update the metadata is no longer supported. To update the metadata from the command line, you can use a response file with Configuration Wizard.

As an alternative to stepping through each page of the Configuration Wizard for each project source that needs to be updated, you can create a response file with the update information and use that response file with the Configuration Wizard to automatically update your metadata. For more information about using a response file to update the metadata, see Using a Response File with Configuration Wizard.

If you do not have access to the GUI mode of Configuration Wizard, you must use a response file to update your metadata.

Prerequisites for Updating a MicroStrategy Metadata

Before attempting to update your MicroStrategy metadata to the latest version, make sure you meet the following prerequisites:

- The metadata update process can be executed only by a MicroStrategy user who is either a member of the System Administrators user group, or is the out-of-the-box Administrator user. Having all administrative privileges is not sufficient: the user must be a member of the System Administrators user group.

- The project's metadata version from which you update must be older than, or the same as, the version of the machine where Configuration Wizard is installed that will be performing the metadata update An older version of
the product is not able to perform an update on a newer version of metadata.

- The language settings of the client, project, and Intelligence Server must all be the same.

- The projects to be updated must not be locked. To unlock a project's metadata, in Developer, from the Administration menu, go to Locking > Unlock Project.

**Database-Specific Prerequisites**

The following prerequisites are specific to the RDBMS that your project metadata is stored in:

- If you are using Oracle for your metadata database, make sure that the Maximum Open Cursors parameter for the database is set to at least 1500.

- If you are using SQL Server for your metadata database, MicroStrategy recommends setting the transaction log to the Simple Recovery Model because the Bulk or Full Recovery Models may fill the transaction log during the upgrade and cause an error.

- If you are using DB2 for your metadata database, make sure that the Dynamic Sections property for the database is set to 999.

**Metadata Update Process**

If you attempt to update a project through the Configuration Wizard and the update fails, the project is locked. You must unlock the project before you attempt to upgrade the project again. To unlock a project, in Developer, from the Administration menu, go to Locking > Unlock Project.
To Make Metadata Updates Using the Configuration Wizard

1. Select the **Upgrade existing environment to MicroStrategy Secure Enterprise** option. Click **Next**.

2. Select the **Intelligence Server components** option. Click **Next**.

3. On the MicroStrategy Authentication page, type the username and password of a MicroStrategy system administrator. Click **Next**.

4. On the metadata Connection page, enter the database password to access the MicroStrategy metadata. Click **Next**.

   The DSN and Login ID are provided based on the server configuration that you did previously.

5. On the Select Components page, select the check boxes for each Intelligence Server you want to upgrade.

6. Under each selected Intelligence Server, select the system components you want to upgrade:

   - **Upgrade metadata repository**: The metadata repository contains the definitions of your MicroStrategy applications and supporting objects. An upgrade of your metadata is required to provide support for all new and updated features in the most recent version of MicroStrategy.

   - **Lean Objects migration**: MicroStrategy 9.3.1 introduced Lean Objects, a new, significantly more compact form of object representation in the metadata tables. Lean Objects are typically 25-50% smaller and will never be larger than the original representation. Migrating existing objects to Lean Objects also allows for faster loading of objects compared to the old representation.

   All new objects are created in this format, and existing objects are converted to the new format as they are saved. Selecting Lean
Objects migration converts all objects in the repository to the new format at once. This option is automatically selected if you select Upgrade metadata repository.

The Lean objects migration modifies all objects in your MicroStrategy projects to use the new object representation. This migration can require a significant amount of time and resources to complete.

Lean Objects migration decreases the amount of disk space used by your objects. However, the migration causes all objects in your projects to be resaved. Depending on your database transaction logging policy, this may cause a temporary increase in the size of your metadata database.

- **Update privileges**: Updating the privileges ensures that users have access to the same functionality that they had in previous versions of MicroStrategy.

  By default, privileges are upgraded based on the version of your MicroStrategy metadata. However, if you previously upgraded your MicroStrategy metadata and did not upgrade privileges along with the metadata upgrade, you must identify the MicroStrategy version you most recently upgraded your privileges for.

  To do this, to the right of **Update privileges**, click **Advanced**. The Select Privilege Version dialog box opens, with the version of your MicroStrategy metadata selected by default. From the drop-down list, select the version of MicroStrategy that privileges were most recently upgraded for and click **OK**.

- **Migrate History List messages**: Select this check box to migrate your existing History List repository to a new format for improved search results, scalability, and performance. A file-based repository can be migrated to a database-based repository or a hybrid
repository, and a database-based repository can be migrated to a hybrid repository.

- If you select this check box, additional configuration options for this migration are provided later, on the History List Migration page. During the upgrade process this Intelligence Server and any other Intelligence Servers in the cluster are stopped and restarted.

- If you are using a file-based History List repository and want to continue using that format, clear this check box. The repository is updated, but its format is not affected, when you restart the Intelligence Server after the upgrade.

- If you are using a database-based or hybrid History List repository and want to continue using the same type of repository, for information about upgrading the History List database, see 2. Update the Metadata.

- **Update Distribution Services Objects**: Select this check box to upgrade Distribution Services subscription objects. This update is required to ensure that your Distribution Services subscriptions can support the new features and enhancements included in the most recent release. If you select this option, ensure that the metadata repository has been upgraded or is selected to be upgraded as part of this update.

- **Database Instance Update**: Select this checkbox to update any MDX database instances which currently use JCO 2.x to JCO 3.x. As of MicroStrategy 10.x the Java Virtual Machine used does not support JCO 2.x. This option will only affect MDX database instances which are set to use JCO 2.x.

7. Select each project you want to update. Only projects that are loaded on Intelligence Server can be updated. Project updates can include any of the following options:
• **Execute project logical upgrade**: Updates the project to the most recent version of MicroStrategy. This option also updates the report, document, and dossier definitions to improve the performance and reduce the memory usage and storage requirements of reports and documents that were created with a previous version of MicroStrategy.

  Be aware of the following:

  • If this update has already been completed, this option is named **Re-execute project logical upgrade**.

  • This upgrade is required to execute any additional project upgrade options.

  • Updated Report Services documents cannot be run in previous versions of MicroStrategy.

  • Depending on the number of reports and documents in the project, this update may take significant time.

• **Lean Objects migration**: Selecting Lean Objects migration converts all objects in this project to the new format at once.

  The Lean objects migration modifies all objects in this project to use the new object representation. This migration can require significant time and resources to complete.

  Lean Objects migration decreases the amount of disk space used by your objects. However, the migration causes all objects in this project to be resaved. Depending on your database transaction logging policy, this may cause a temporary increase in the size of your metadata database.

• **Update Schedules**: Makes previous subscriptions viewable in MicroStrategy 2020.
- **Update MDX Source Objects**: Updates the MDX data source objects (MDX Cubes) that were created in earlier versions of MicroStrategy to take advantage of improved performance.

  By default, if you update multiple projects and the update fails for one project, the update process continues for other projects. This allows you to complete all possible updates and then review any errors. To abort the update process for other projects when one project fails, clear the **Update the remaining projects, even when one project update fails** check box.

  Depending on the number of MDX source objects in the project, this update may take significant time.

- **Update OLAP Services Cube caches to PRIME**: Select this check box to update all Intelligent Cubes to the PRIME architecture. This architecture can improve the performance of publishing Intelligent Cube results through the use of parallel processing and other optimizations. This update is applied to all Intelligent Cubes for the project.

- **Update data import cubes (definition and cache) to PRIME**: Select this check box to update all Data Import datasets to the PRIME architecture. This architecture can improve the performance of publishing Data Import results through the use of parallel processing and other optimizations. This update is applied to data sets included in your project through the use of Data Import.

  It is easier to update OLAP Services Cube caches and data import cubes to PRIME during the upgrade process. If you do not update them now, each cube is updated when it is used. Doing the upgrade all at once ensures consistent performance for end users.

8. Click **Next**.
9. If you selected the **Migrate History List messages** option, on the History List Target Database page, specify the necessary information for the History List database.

10. Click **Next**.

11. On the Summary page, review your upgrade choices and click **Finish**.

3. History List Database Repository

If you are using a database-based History List, when upgrading to MicroStrategy 2020 you must upgrade the History List database tables. MicroStrategy recommends using the Configuration Wizard to upgrade the History List repository.

To Upgrade the History List Repository

1. In Windows, go to **Start > All Programs > MicroStrategy Tools > Configuration Wizard**. The Configuration Wizard opens.

2. Select **Upgrade existing environment to MicroStrategy Secure Enterprise** and click **Next**.

3. Select **History List Repositories** and click **Next**.

4. Select the DSN for the History List database, and specify the login information.

5. To upgrade existing History List messages to the current format, select the **Copy History List content** check box.

   If this check box is cleared, existing History List messages are not upgraded. The new tables required to support History List messages are still created.

6. If you select to upgrade your existing History List messages, you can
also select the **Compress data** check box to compress all existing History List messages. This compression can improve the performance of using History List messages. However, the compression of your History List messages can require significant system resources during the upgrade.

To avoid the overhead of compressing your existing History List messages, clear the **Compress data** check box. Existing History List messages are not modified, but any newly created History List messages are created with the new compressed format.

7. Click **Next**.

8. Review the available information, and click **Finish**.

4. MicroStrategy Web and Mobile Server

Once you have upgraded Intelligence Server and updated the project metadata, upgrade your MicroStrategy Web and Mobile Server installations.

**.NET:**

Upgrading a Windows machine (IIS) with MicroStrategy Web and Mobile Server directly from an older version to a newer version upgrades both deployments automatically. In-place upgrades require no additional steps. To upgrade MicroStrategy Web and Mobile Server, Administrators should follow the detailed instructions in [Deploying with IIS (Windows)].

**J2EE:**

Upgrading MicroStrategy Web and/or Mobile for JSP, whether it's on Windows or Linux, only creates a new `MicroStrategy.war` and/or `MicroStrategyMobile.war` file. Deployments are not done by the MicroStrategy Installer and must be done by the administrator on a supported application server.

Before upgrading MicroStrategy Web and Mobile Server, be aware of the following considerations:
• Note your current Web customizations and review the information in MicroStrategy Web Customizations to learn how to successfully migrate these customizations.

• If you have created any mobile device configurations or saved any images from the MicroStrategy Photo Uploader widget, those configurations and images are deleted during the upgrade unless you manually back them up before the upgrade and restore them after the upgrade.

• To back up your configuration files, make a copy of the directory `<MicroStrategyMobileServer>/WEB-INF/xml/mobile/` and all its contents, where `<MicroStrategyMobileServer>` is the installation location of MicroStrategy Mobile Server. By default, this location is `C:\Program Files (x86)\MicroStrategy\Mobile Server ASPx` or `\Mobile Server JSP`. Then, after you have upgraded Mobile Server, copy the contents of the backed-up directory to the same corresponding location in the new Mobile Server deployment.

• Photo Uploader images are stored in MicroStrategy Mobile Server. To back up your Photo Uploader images, make a copy of the directory `<MicroStrategyMobileServer>/WEB-INF/SavedImages/` and all its contents, where `<MicroStrategyMobileServer>` is the installation location of MicroStrategy Mobile Server. By default, this location is `C:\Program Files (x86)\MicroStrategy\Mobile Server ASPx` or `\Mobile Server JSP`.


• After upgrading MicroStrategy Web or Mobile Server, restart the web and application servers along with clearing the web and application server caches.

For steps specific to your J2EE environment, see:
4.1 MicroStrategy Web Customizations

MicroStrategy Web customizations are developed as plugins that are stored in the plugins folder of the environment. When upgrading to a newer version of MicroStrategy Web, the plugin upgrade differs depending on whether you are using a .NET environment on Windows or a J2EE environment on Windows or Linux.

.NET Environment

When you upgrade MicroStrategy Web in a .NET environment (IIS), the existing plugins are automatically copied to the plugins folder of the new installation directory.

J2EE Environment

When you upgrade MicroStrategy Web in a J2EE environment, you must manually copy your existing plugins to the plugins folder in your new deployment folder.

1. Make a copy of your existing plugins.
3. Copy your existing plugins to the new plugins folder in deployment folder.

For information about the plugin structure used for MicroStrategy Web customizations, refer to MicroStrategy Web SDK in the MicroStrategy Developer Library.
5. MicroStrategy Library

In MicroStrategy 10.9 through 11.0, MicroStrategy Library was automatically deployed in Tomcat and the MicroStrategyLibrary.war file installed in the <INSTALL_PATH>\LibraryWebMobile directory. MicroStrategy 2019 and above requires administrators to perform the upgrade for their Library deployment.

You must manually deploy a new Library WAR File after upgrading MicroStrategy Library.

This section contains high level steps and considerations for performing an upgrade or migrating MicroStrategy Library to a new environment.

Windows Environments

In MicroStrategy 10.9 to 11.0, MicroStrategy Library is automatically deployed in Tomcat and the Library.war file is installed in the <INSTALL_PATH>\LibraryWebMobile directory. Starting in MicroStrategy 2019, administrators can choose Tomcat as an option under "Other Components" to deploy Library to Tomcat, or deselect it so that only the WAR file is created in the directory referenced above.

Perform the following steps when doing a direct upgrade of MicroStrategy Library 10.9 - 11.0 to MicroStrategy 2020.

1. Within the Tomcat directory, the following files are backed up:
   - webapps folder
   - server.xml
   - tomcat-users.xml

2. Uninstall old Tomcat.


4. All backed up files are restored to the new Tomcat installation.
5. Start Tomcat.

6. The new Library.war file is only installed in `<INSTALL_PATH>\LibraryWebMobile`.

Deploying the New WAR File

A new Library WAR File must be manually deployed after upgrading MicroStrategy Library, even between service packs.

You should backup the `webapps` folder as they are overwritten when deploying new `.war` files.

Use the following steps to deploy a new WAR file for both Windows and Linux.

1. Use the MicroStrategy installer to create a new MicroStrategy 2020 Library.war file by doing a direct upgrade or new installation.

2. Redeploy the new Library.war file from the Library install directory to the J2EE server being used. See the list of supported servers.

3. Migrate your customizations to the new environment, such as:
   - `.../webapps/MicroStrategyLibrary/images/`
   - `.../webapps/MicroStrategyLibrary/plugins/`
   - `.../webapps/MicroStrategyLibrary/WEB-INF/classes/auth/`
     - Customized authentication configuration, such as SAML authentication
   - `.../webapps/MicroStrategyLibrary/WEB-INF/classes/config/configOverride.properties`
   - Make sure MicroStrategy Library is pointing to the correct Intelligence Server and Collaboration Server with correct
authentication setting by going to the Library Administration Control Panel.

- \(.../webapps/MicroStrategyLibrary/WEB-INF/classes/logback.xml\)

- Any other customizations including SSL configurations

4. Verify that MicroStrategy Library functionality including collaboration, sharing, exporting, printing, and all other features are working properly. Ensure dossiers and documents are appearing.

The Collaboration Server is a separate component of MicroStrategy that is upgraded independently of Library. See Upgrade MicroStrategy Collaboration for more information.

6. MicroStrategy Collaboration

MicroStrategy Collaboration has been enhanced to use a central data repository installed as part of the MicroStrategy Platform. Starting in MicroStrategy 2020 the Collaboration Server constructs the connection string to the MicroStrategy Data Repository during runtime. After performing the upgrade, Collaboration Server will only support connections to the MicroStrategy Data Repository.

After upgrading Collaboration Server, the previous data repository remains unchanged but it will not be used by the new version of Collaboration Server. Any comments stored in the previous data repository will need to be moved to the new repository to be accessible.

Migrating Comments to the MicroStrategy Data Repository

The Collaboration Server Administration Tool provides a way for system administrators to migrate comments from the previous data repository to the new data repository. The Collaboration Administration Page will display a warning message when it detects that a previous data repository was used by the Collaboration Server before the upgrade.
The Collaboration Server Administration Tool is a console application and will connect to the data repository using the same configuration as the Collaboration Server so no user input is required to connect. The tool will provide feedback as data is being migrated and will show a message when the migration is complete.

In order to recover collaboration messages that were stored in the previous version of the MicroStrategy Platform please follow these steps:

1. Click **Start > Windows Administrative Tools > Services**.


3. Start the service Mongo DB.

4. Run the Data Migration Command:
   
   - Enter the three file locations in the following format:
     
     ```shell
     "$ <PATH_TO_NODE>" "$ <PATH_TO_ADMIN_TOOL_FILE>" "$ <PATH_TO_CONFIG_FILE>"
     ```
     
     ```bash
     "C:\Program Files\Common Files\MicroStrategy\nodejs\node.exe" 
     "C:\Program Files\MicroStrategy\Collaboration Server\node_modules\mstr-collab-svc\admintool.js" 
     "C:\Program Files\MicroStrategy\Collaboration Server\config.json"
     ```
   
   - Use the **migrate** command to run the data migration:
     
     ```bash
     cmd> migrate
     ```
   
   - Enter **quit** to close the Collaboration Server Administration Tool.

5. Stop the service Mongo DB.


The Collaboration Admin Page will stop displaying the warning message at this point.
7. Updating the Enterprise Manager Project

Upgrading from MicroStrategy 9 to 2020 includes major enhancements and updates to the Enterprise Manager project; whereas upgrading from MicroStrategy 10 to 2020 does not have major enhancements and upgrades to the Enterprise Manager project. In both scenarios, you should use the latest versions of Enterprise Manager projects with MicroStrategy 2020.

To retain customizations created in previous Enterprise Manager projects, make separate copies of these customized copies of the out-of-the-box objects, or as brand-new objects. Any modifications you made to the out-of-the-box objects for an Enterprise Manager project are replaced as part of the upgrade process.

- To upgrade your Enterprise Manager statistics and warehouse, ensure your Enterprise Manager Data Loader service is pointing to your Statistics and Enterprise Manager Repository. For more information, see KB483298.

- Use the Project Duplication Wizard to make a backup of your existing Enterprise Manager project. For detailed information about using Project Duplication, see the Managing Your Projects chapter of the System Administration Guide.

1. Open Configuration Wizard.

2. Select Upgrade existing environment to MicroStrategy Secure Enterprise, and click Next.

3. Select Upgrade Enterprise Manager Project, and click Next.

4. Provide the following information:

   - **User Name**: Type the MicroStrategy user name that can access and administer the Enterprise Manager project.

   - **Password**: Type the password for the MicroStrategy user that can access and administer the Enterprise Manager project.
5. If you have an Enterprise Manager project package file (.mmp) to use instead of the default file, click **Advanced >>**. For the **Package location** field, click ... (the Browse button) to navigate to and select the package file.

6. Click **Next**.

7. Provide the following information:
   - **DSN**: Select the data source name for your statistics repository or 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.

8. Click **Next**.

9. Review the summary information.

   You can click **Save** to save the configuration as a response (.ini) file to upgrade your Enterprise Manager projects on other systems or to run silent configurations at a later time. For information on running the Configuration Wizard with a response file, see *Using a Response File with Configuration Wizard*.

10. Click **Finish**.

8. **Upgrade the Platform Analytics Project**

As of 2019, you can upgrade your Platform Analytics project in the metadata of your connected Intelligence Server. Upgrading the project is recommended with each platform and update release in order to brings in the latest dossiers, attributes, metrics and reporting optimizations to the Platform Analytics project.
1. Open Configuration Wizard.

2. Select Upgrade existing environment to MicroStrategy Secure Enterprise, and click Next.

3. Select Upgrade Platform Analytics Project, and click Next.

4. Provide the following information:
   - **User Name**: Enter the MicroStrategy user name that can access the Intelligence Server.
     
     If this is your first time connecting to the MicroStrategy Intelligence Server, use the user name Administrator without a password.

   - **Password**: Enter the password for the MicroStrategy user that can access the Intelligence Server.

5. Choose the Platform Analytics Repository.

   Select the MySQL/PostgreSQL DSN for the Platform Analytics Repository that was used to create Platform Analytics.

6. Enter your **User Name** and **Password** for the DSN.

7. Click Next.

8. Click Apply. The Configuration Wizard automatically applies the following configuration files:
   - PlatformAnalyticsConfigurationNew scp
   - PlatformAnalyticsConfigurationNew_PostgreSQL scp
   - PlatformAnalyticsConfigurationUpgrade scp
   - PlatformAnalyticsConfigurationUpgrade_PostgreSQL scp
9. If an error appears about being unable to automatically apply project settings to Platform Analytics, you must manually update the project settings. For instructions, see Configure the Platform Analytics Project.

9. Upgrade the Platform Analytics Repository

After installing Platform Analytics, there is a Platform Analytics Repository (Platform Analytics warehouse table) where you can create the Platform Analytics project on your local machine. Upgrading the repository is required with each platform and update release in order to benefit from Platform Analytics warehouse new features, defect fixes and database structure optimizations.

The Configuration Wizard provides the following options:

- **Host**: Type the host name of the Platform Analytics warehouse. By default, this is set to the last successful connection value.

- **Port**: Type the port number of the Platform Analytics warehouse. By default, this is set to the last successful connection value.

- **User Name**: Type the user name for the Platform Analytics warehouse. By default, this is set to the value from PAConsumerConfig.yaml file

- **Password**: Type the password for the Platform Analytics warehouse user.

Depending on the warehouse type you choose for the Host and Port, you must set the parameter `whDbType` to either "postgresql" or "mysql" in the PAConsumerConfig.yaml file.

The default path is:

- **Linux**: `/opt/MicroStrategy/PlatformAnalytics/Conf`

- **Windows**: `C:\Program Files (x86)\MicroStrategy\Platform Analytics\conf`

Click **Next** to proceed.
You can also update the Platform Analytics repository using the Configuration Wizard in interactive mode.

How to Update the Repository in Interactive Mode

To update the Platform Analytics repository using the Configuration Wizard in interactive mode on Windows:

1. In a Windows console, enter one of the following commands:
   - For 64-bit, enter `MACfgWiz_64`.
   - For 32-bit, enter `MACfgWiz`.
2. Click Enter.
3. Type 2 and click Enter to create a new `response.ini` file.
4. Type 5 and click Enter to upgrade your existing environment to MicroStrategy Analytics Enterprise.
5. Type 3 and click Enter to upgrade your Platform Analytics repository.
6. Enter your Platform Analytics warehouse database credentials. By default, the server name, port number, and user name are set to the last successful connection value.

   If you did not change the values, leave as default. The default password can be found at `C:\Program Files (x86)\Common Files\MicroStrategy\express_password.txt`

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 click **Enter**. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

8. Type **Y** and click **Enter** to run the configuration.

To update the Platform Analytics repository using the Configuration Wizard in interactive mode on Linux:

1. In a Linux console window, browse to **HOME_PATH** where **HOME_PATH** is the specified home directory during installation.

2. Browse to the **bin** directory.

3. At the command prompt, type **mstrcfgwiz-editor**, then click **Enter**. The Configuration Wizard opens in command line mode.

4. Click **Enter**.

5. Type **2** and click **Enter** to create a new **response.ini** file.

6. Type **5** and click **Enter** to upgrade your existing environment to MicroStrategy Analytics Enterprise.

7. Type **3** and click Enter to upgrade your Platform Analytics repository.

8. Enter your Platform Analytics warehouse database credentials. By default, the server name, port number, and user name are set to the last successful connection value.

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 click **Enter**. The response.ini file is generated, and you
are prompted whether to run the configuration immediately.

10. Type \texttt{Y} and click \texttt{Enter} to run the configuration.

10. Activate Your Installation

All MicroStrategy 2020 installations must be activated within 30 days. Once you receive the activation code you will need to enter it into MicroStrategy License Manager.

Instructions

1. Open MicroStrategy License Manager from \texttt{Start menu} > \texttt{Programs} > \texttt{MicroStrategy Tools} > \texttt{License Manager}.

2. Go to the \texttt{License Administration} tab > \texttt{Activate Server Installation} and click \texttt{Next}.

3. Select the \texttt{Server Activation using Activation Code} option and enter your Activation Code in the text field. Click \texttt{Next}.

4. Click \texttt{OK}.

Performing an In-Place Upgrade on a Linux Deployment

This section covers the procedure for directly upgrading your currently deployed Linux environment to MicroStrategy 2020. Carefully review the \textit{Planning Your Upgrade} section before proceeding with your upgrade.

1. Upgrade Preparation

Before you begin the upgrade to MicroStrategy 2020 take care of the following items:

- Log in as \texttt{root} user.
- Backup the current configuration files for Intelligence Server, Web,
Mobile, Library, and Collaboration.

List of Files:

Intelligence Server

- MSIReg files
- odbcinst.ini
- odbc.ini

MicroStrategy Library

- ../webapps/MicroStrategyLibrary/images/
- ../webapps/MicroStrategyLibrary/plugins/
- ../webapps/MicroStrategyLibrary/WEB-INF/classes/auth/
- ../webapps/MicroStrategyLibrary/WEB-INF/classes/config/configOverride.properties
- ../webapps/MicroStrategyLibrary/WEB-INF/classes/logback.xml

- Any other customizations including SSL configurations

MicroStrategy Collaboration

When upgrading the Collaboration Server, the MicroStrategy Installer preserves the config.json file for the service; however, it does not back up the notification config.json file. If you have modified any notification properties, you need to backup the following and restore it once the upgrade is complete:

Backup Collaboration Server Notifications
When upgrading the Collaboration Server, the MicroStrategy Installer preserves the `config.json` file for the service; however, it does not back up the notification `config.json` file. If you have modified any notification properties, you need to backup the following and restore it once the upgrade is complete:

Back up the Collaboration Server notification files in the following directory:

```
<INSTALL_PATH>\MicroStrategy\Collaboration Server\node_modules\mstr-collab-svc\pluginConfig\dossier
```

List of files:

- `config.json`
- `email-Invite.template`
- `email-Mention.template`

Backup Collaboration Store

The Collaboration Server utilizes MongoDB as its repository database. MongoDB has a built-in command for database backup, transfer, and restoration.

MicroStrategy recommends that the MongoDB database be backed up on a regular basis.

Gather the following information from the `dburl` field in the `config.json` file from each server:

- Username
- Password
- Collaboration database name
1. Stop the Collaboration Server and the MongoDB service.

2. Execute the following command from the destination server:

```bash
mongodump --host <source_server_name> --db <source_db_name> --username <source_username> --password <source_password> --authenticationDatabase admin --excludeCollection sessions --archive |
mongorestore --username <target_username> --password <target_password> --authenticationDatabase admin --drop --archive
```

3. Restart the Collaboration Service and the MongoDB service.

- Send an email to your users to inform that Intelligence Server and Web Server will be down during the upgrade process.
- Halt any monitoring alerts in case false alarm triggered during upgrades.
- Check the MicroStrategy Web Administration page and delete any defined Trust Relationships.
- Restart all services for the machines to be upgraded.
- Execute `service mstr stop` to stop Tomcat, MicroStrategy Listener, and Intelligence Server.
- Unmount all current mounts to any MicroStrategy deployment.

Unmounting is necessary to prevent the installer from attempting to remove other items in the mounted locations such as the image shares.

- To view the related mounts and locations, run:
  ```bash
  mount | grep MicroStrategy
  ```
- To unmount each folder, run `umount /<path>/<to>/<folder>`. For example:
  ```bash
  umount /opt/mstr/MicroStrategy/install/images
  ```
• Stop puppet if the service is running.

Use the following commands:

• **Stop puppet:** `service puppet stop`

• **Get process id for any additional puppet processes:**
  
  ```
  ps -ef | grep puppet
  ```

• **Kill any remaining processes:** `kill -9 <processID>`

• Remove MicroStrategy directory and .war files from tomcat webapps and work folders.
  
  1. `cd /opt/apache/tomcat/latest/webapps/`
  2. `rm -rf MicroStrategy*`
  3. `cd /opt/apache/tomcat/latest/work/Catalina/localhost`
  4. `rm -rf MicroStrategy*`

• **Use df -h to check that at least 5GB of disk space is available under /opt/mstr**

2. **Install MicroStrategy 2020**

The MicroStrategy Installation Wizard can be used in either in graphical user interface (GUI) mode or in command line mode for Linux.

1. Navigate to your MicroStrategy 2020 download folder and then `QueryReportingAnalysis_Linux`.

2. Run the Installation Wizard:

   • **For GUI mode:** `./setup.sh`
   
   • **For command line mode:** `./setup.sh -console`

   *After each step of the install process click Next, or type 1 and press*
Enter, to proceed.

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

3. Step through each of the Installation Wizard windows/prompts:

1. Accept the MicroStrategy License Agreement.
2. Update your license key.
3. Review the list of products included in your license.
4. Review the products to be installed.
   
   If your license includes new products or services select them for installation in this window.
5. The installer displays any running services that need to be stopped. Click Yes to stop them now.
7. Server activation information
8. Enter your contact information for server activation.
9. Choose to have an activation code sent you to now or at a later date.
10. Review the list of components selected for installation.
   
   Choose if you would like the installer to automatically reboot the machine if necessary.
11. Click Install to begin.

3. Upgrade the Drivers for Your Data Source Connections

MicroStrategy 2020 requires 64-bit drivers for all data source connections.

Upgrading from versions prior to MicroStrategy 10.4...
data source using a MicroStrategy-branded driver, the drivers are upgraded from 32-bit to 64-bit drivers automatically when Intelligence Server is upgraded. You must manually update the MicroStrategy odbc.ini configuration file to point to the location for the 64-bit driver.

All MicroStrategy-branded drivers are upgraded during the installation process. Simply replace the newly installed odbc.ini with the backup taken before the upgrade.

If you connect to your data source using a driver provided by a third-party vendor, you must install the 64-bit version of the driver from your third-party vendor, and then create a new DSN that uses that driver to connect to your data source. You must also update the MicroStrategy odbc.ini configuration file.

After you upgrade the drivers, continue the upgrade process with 4. Connect the Intelligence Server to the Metadata.

4. Connect the Intelligence Server to the Metadata

Use the Configuration Wizard to establish a connection between Intelligence Server and your metadata repository.

Configure Metadata Connection

1. Run the MicroStrategy Configuration Wizard:
   
   cd /opt/mstr/Microstrategy/bin
   .\mstrcfgwiz

2. Select Configure Intelligence Server, and click Next.

3. From the DSN drop-down list, select the DSN used to connect to the metadata repository. Enter the User Name and Password for the database.

4. Click Next.
5. Select **Use the selected Server Definition as active.**

6. Choose the server definition to use from the **Existing Server Definitions** pane.

7. Click **Next.**

Define the Intelligence Server port number and other settings

1. Define the Intelligence Server settings described below:

   - **Port number**: You can use the default port number (34952) or specify another port number. The port number is how a server process identifies itself on the machine on which it is running. If the port number is used by another process, such as in a shared environment, specify an available port number.

   - **REST port number**: You can use the default port number (34962) or specify another port number for the REST API Server inside Intelligence Server. This port number should not be same with the Intelligence Server port number.

     You can configure REST API Server logging with the Diagnostics and Performance Logging Tool. Select the **Performance Configuration** tab, and find the dispatcher **REST Trace** under component **Network Classes**. For more information, see the Configuring What is Logged section in the **System Administration Guide**.

   - **Register Intelligence Server as a Service**: Select this check box to register Intelligence Server as a service.

   - **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.

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

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

2. Click **Next**.

3. Secure socket layer (SSL) protocol encrypts communications between Intelligence Server and Developer:

   **Configure SSL**: This option specifies whether to enable Intelligence Server and Developer to communicate using the SSL protocol. Clear this check box to disable the use of the SSL protocol for Intelligence Server and Developer communications. This option also applies SSL protocol to the REST port number.

   When you select to enable the SSL protocol, you must provide the following information:

   • **Certificate**: The SSL certificate file you created for Intelligence Server. Click the browse button to navigate to and select the certificate file.

   • **Key**: The private key file you created while requesting the certificate for Intelligence Server. Click the browse button to navigate to and select the private key file.
• **Password**: The password that you used while creating the private key for the SSL certificate.

• **SSL Port**: The port number to use for SSL access. By default, the port is 39321.

To enable SSL protocol communication in Developer, you must use the Project Source Editor. For steps to complete the other tasks required to enable SSL protocol communications, refer to the System Administration Guide.

**Configuring port requires Client Certificate**: If selected, the SSL communications with client certificate verification will be configured in the Intelligence Server, but does not apply to REST port number. Provide the following information:

• **SSL Port**: The port number the Intelligence Server will use for SSL communications with client certificate verification.

• **Truststore**: The location to the client certificate truststore.

4. Click **Next**.

5. To specify the default statistics repository for the local Intelligence Server and 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)**: Basic statistics for new projects and any projects that are not currently logging statistics will be enabled. You can alter the settings in Project Configuration Editor in MicroStrategy Developer to:
  - Enable additional statistics for a project.
  - Enable basic and additional statistics for a project if you cleared this check box.

6. Enable and configure the Messaging Services host and port settings.

   **Messaging Services Host(s)/Port(s)**: Provide host name or IP address of Messaging Services.

   Format for this setting should be: `server1.acme.com:9092`

7. Click **Next**.

8. Review the summary information and click **Save**.

9. Click **Finish**.

5. Update the Metadata

After upgrading the Intelligence Server, update the metadata by performing the procedure that follows. This procedure creates and updates metadata
tables to support new features available in MicroStrategy 2020.

Updating the metadata ensure compatibility between your pre-existing objects and the MicroStrategy clients. You can also take advantage of new configuration settings exposed through the client applications, functionality, and objects available only with the updated version of metadata.

During the metadata update process, the Configuration Wizard generates additional SQL for tasks such as managing indexes and primary keys. To view this SQL during the update process, on the Summary Page of the Configuration Wizard, in the Summary pane, click SQL Preview. In addition, once you have completed the update process, the generated SQL is saved in the MicroStrategy common files directory. The file name is OriginalScript_generated.sql, where OriginalScript is the name of the original SQL update script. For example, if you are updating an Oracle database, the original SQL update script is mdorcl.sql and the generated SQL file is mdorcl_generated.sql.

Be aware of the following:

- If you have made any changes to the privileges assigned to the out-of-the-box MicroStrategy user groups and security roles, updating the metadata may overwrite those changes. In particular, if you have made any changes to groups that use any privileges mentioned in the List of updated privileges, updating the metadata will overwrite those changes.

- For information about viewing the Configuration Wizard log file if an error occurs, see Resolving Problems Encountered During the Upgrade.

- The MDUpdate command line utility that was formerly used to update the metadata is no longer supported. To update the metadata from the command line, you can use a response file with Configuration Wizard.

As an alternative to stepping through each page of the Configuration Wizard for each project source that needs to be updated, you can create a response file with the update information and use that response file with the Configuration Wizard to automatically update your metadata. For more
information about using a response file to update the metadata, see *Using a Response File with Configuration Wizard*.

If you do not have access to the GUI mode of Configuration Wizard, you must use a response file to update your metadata.

Prerequisites for Updating a MicroStrategy Metadata

Before attempting to update your MicroStrategy metadata to the latest version, make sure you meet the following prerequisites:

- The metadata update process can be executed only by a MicroStrategy user who is either a member of the System Administrators user group, or is the out-of-the-box Administrator user. Having all administrative privileges is not sufficient: the user must be a member of the System Administrators user group.

- The project's metadata version from which you update must be older than, or the same as, the version of the machine where Configuration Wizard is installed that will be performing the metadata update. An older version of the product is not able to perform an update on a newer version of metadata.

- The language settings of the client, project, and Intelligence Server must all be the same.

- The projects to be updated must not be locked. To unlock a project's metadata, in Developer, from the *Administration* menu, go to *Locking > Unlock Project*.

Database-Specific Prerequisites

The following prerequisites are specific to the RDBMS that your project metadata is stored in:

- If you are using Oracle for your metadata database, make sure that the Maximum Open Cursors parameter for the database is set to at least
If you are using SQL Server for your metadata database, MicroStrategy recommends setting the transaction log to the Simple Recovery Model because the Bulk or Full Recovery Models may fill the transaction log during the upgrade and cause an error.

If you are using DB2 for your metadata database, make sure that the Dynamic Sections property for the database is set to 999.

To Make Metadata Updates Using the Configuration Wizard

1. Select the **Upgrade existing environment to MicroStrategy Secure Enterprise** option. Click **Next**.

2. Select the **Intelligence Server components** option. Click **Next**.

3. On the MicroStrategy Authentication page, type the username and password of a MicroStrategy system administrator. Click **Next**.

4. On the Metadata Connection page, enter the database password to access the MicroStrategy metadata. Click **Next**.

   The DSN and Login ID are provided based on the server configuration that you did previously.

5. On the Select Components page, select the check boxes for each Intelligence Server you want to upgrade.

6. Under each selected Intelligence Server, select the system components you want to upgrade:

   - **Upgrade metadata repository**: The metadata repository contains the definitions of your MicroStrategy applications and supporting objects. An upgrade of your metadata is required to provide support for all new and updated features in the most recent version of MicroStrategy.
• **Lean Objects migration**: Lean Objects are a significantly more compact form of object representation in the metadata tables. Lean Objects are typically 25-50% smaller and will never be larger than the original representation. Migrating existing objects to Lean Objects also allows for faster loading of objects compared to the old representation.

All new objects are created in this format, and existing objects are converted to the new format as they are saved. Selecting Lean Objects migration converts all objects in the repository to the new format at once. This option is automatically selected if you select Upgrade metadata repository.

The Lean objects migration modifies all objects in your MicroStrategy projects to use the new object representation. This migration can require a significant amount of time and resources to complete.

Lean Objects migration decreases the amount of disk space used by your objects. However, the migration causes all objects in your projects to be resaved. Depending on your database transaction logging policy, this may cause a temporary increase in the size of your metadata database.

• **Update privileges**: Updating the privileges ensures that users have access to the same functionality that they had in previous versions of MicroStrategy.

By default, privileges are upgraded based on the version of your MicroStrategy metadata. However, if you previously upgraded your MicroStrategy metadata and did not upgrade privileges along with the metadata upgrade, you must identify the MicroStrategy version you most recently upgraded your privileges for.

To do this, to the right of **Update privileges**, click **Advanced**. The Select Privilege Version dialog box opens, with the version of your
MicroStrategy metadata selected by default. From the drop-down list, select the version of MicroStrategy that privileges were most recently upgraded for and click **OK**.

- **Migrate History List messages**: Select this check box to migrate your existing History List repository to a new format for improved search results, scalability, and performance. A file-based repository can be migrated to a database-based repository or a hybrid repository, and a database-based repository can be migrated to a hybrid repository.

- If you select this check box, additional configuration options for this migration are provided later, on the History List Migration page. During the upgrade process this Intelligence Server and any other Intelligence Servers in the cluster are stopped and restarted.

- If you are using a file-based History List repository and want to continue using that format, clear this check box. The repository is updated, but its format is not affected, when you restart the Intelligence Server after the upgrade.

- If you are using a database-based or hybrid History List repository and want to continue using the same type of repository, for information about upgrading the History List database, see **5. Update the Metadata**.

- **Update Distribution Services Objects**: Select this check box to upgrade Distribution Services subscription objects. This update is required to ensure that your Distribution Services subscriptions can support the new features and enhancements included in the most recent release. If you select this option, ensure that the metadata repository has been upgraded or is selected to be upgraded as part of this update.
- **Database Instance Update**: Select this checkbox to update any MDX database instances which currently use JCO 2.x to JCO 3.x. As of MicroStrategy 10.x the Java Virtual Machine used does not support JCO 2.x. This option will only affect MDX database instances which are set to use JCO 2.x.

7. Select each project you want to update. Only projects that are loaded on Intelligence Server can be updated. Project updates can include any of the following options:

- **Execute project logical upgrade**: Updates the project to the most recent version of MicroStrategy. This option also updates the report, document, and dossier definitions to improve the performance and reduce the memory usage and storage requirements of reports and documents that were created with a previous version of MicroStrategy.

  Be aware of the following:

  - If this update has already been completed, this option is named *Re-execute project logical upgrade*.

  - This upgrade is required to execute any additional project upgrade options.

  - Updated Report Services documents cannot be run in previous versions of MicroStrategy.

  - Depending on the number of reports and documents in the project, this update may take significant time.

- **Lean Objects migration**: Selecting Lean Objects migration converts all objects in this project to the new format at once.

  The Lean objects migration modifies all objects in this project to use the new object representation. This migration can require significant time and resources to complete.
Lean Objects migration decreases the amount of disk space used by your objects. However, the migration causes all objects in this project to be resaved. Depending on your database transaction logging policy, this may cause a temporary increase in the size of your metadata database.

- **Update Schedules**: Makes previous subscriptions viewable in MicroStrategy 2020.

- **Update MDX Source Objects**: Updates the MDX data source objects (MDX Cubes) that were created in earlier versions of MicroStrategy to take advantage of improved performance.

  By default, if you update multiple projects and the update fails for one project, the update process continues for other projects. This allows you to complete all possible updates and then review any errors. To abort the update process for other projects when one project fails, clear the **Update the remaining projects, even when one project update fails** check box.

  Depending on the number of MDX source objects in the project, this update may take significant time.

- **Update OLAP Services Cube caches to PRIME**: Select this check box to update all Intelligent Cubes to the PRIME architecture. This architecture can improve the performance of publishing Intelligent Cube results through the use of parallel processing and other optimizations. This update is applied to all Intelligent Cubes for the project.

- **Update data import cubes (definition and cache) to PRIME**: Select this check box to update all Data Import datasets to the PRIME architecture. This architecture can improve the performance of publishing Data Import results through the use of parallel processing
and other optimizations. This update is applied to data sets included in your project through the use of Data Import.

It is easier to update OLAP Services Cube caches and data import cubes to PRIME during the upgrade process. If you do not update them now, each cube is updated when it is used. Doing the upgrade all at once ensures consistent performance for end users.

8. Click **Next**.

9. If you selected the **Migrate History List messages** option, on the History List Target Database page, specify the necessary information for the History List database.

10. Click **Next**.

11. On the Summary page, review your upgrade choices and click **Finish**.

### 6. History List Database Repository

If you are using a database-based History List, when upgrading to MicroStrategy 2020 you must upgrade the History List database tables.

**To Upgrade the History List Repository**

1. From the Configuration Wizard Welcome page select **Upgrade existing environment to MicroStrategy Secure Enterprise** and click **Next**.

2. Select **History List Repositories** and click **Next**.

3. Select the DSN for the History List database, and specify the login information.

4. To upgrade existing History List messages to the current format, select the **Copy History List content** check box.
If this check box is cleared, existing History List messages are not upgraded. The new tables required to support History List messages are still created.

5. If you select to upgrade your existing History List messages, you can also select the **Compress data** check box to compress all existing History List messages. This compression can improve the performance of using History List messages. However, the compression of your History List messages can require significant system resources during the upgrade.

   To avoid the overhead of compressing your existing History List messages, clear the **Compress data** check box. Existing History List messages are not modified, but any newly created History List messages are created with the new compressed format.

6. Click **Next**.

7. Review the available information, and click **Finish**.

7. **MicroStrategy Web and Mobile Server**

Upgrading MicroStrategy Web and Mobile Server JSP will only create new MicroStrategy.war and MicroStrategyMobile.war files in the installation directory. You will still need to deploy the new .war file to a supported application server.

Before upgrading MicroStrategy Web and Mobile Server, be aware of the following considerations:

- Backup any changed MicroStrategy Web configuration files:
  - `../WEB-INF/xml/config/mapConfig.xml`
  - `../WEB-INF/xml/sys_defaults.properties`
Upgrade Guide

- ../WEB-INF/xml/sys_defaults.xml
- ../WEB-INF/web.xml

- Backup any changed MicroStrategy Mobile configuration files:
  - /WEB-INF/xml/sys_defaults.properties
  - /WEB-INF/xml/sys_defaults.xml
  - /WEB-INF/web.xml

- Backup any mobile device configurations or saved any images from the MicroStrategy Photo Uploader widget, manually back them up before proceeding with your upgrade.
  - /WEB-INF/xml/mobile
  - /WEB-INF/SavedImages

- If you deploy any Web or Mobile customizations and review the following SDK documentation to learn how to successfully migrate these customizations:
  - MicroStrategy Web
  - MicroStrategy Mobile

- If you secure your MicroStrategy Mobile connections through Certificate Server, you must upgrade your Certificate Server to support MicroStrategy 2020 Mobile applications.

For steps specific to your application server, see:

- Deploying with Tomcat (Linux)
- Deploying with Oracle Glassfish Server (Solaris)
- Deploying with WebLogic and Apache (Solaris)
- Deploying with WebSphere and IBM HTTP Server (AIX)
8. MicroStrategy Library

In MicroStrategy 10.9 through 11.0, MicroStrategy Library was automatically deployed in Tomcat and the MicroStrategyLibrary.war file installed in the `<INSTALL_PATH>\LibraryWebMobile` directory. MicroStrategy 2019 and above requires administrators to perform the upgrade for their Library deployment.

You must manually deploy a new Library WAR File after upgrading MicroStrategy Library.

This section contains high level steps and considerations for performing an upgrade or migrating MicroStrategy Library to a new environment.

Linux Environments

In MicroStrategy 10.9 to 11.0, the MicroStrategy installer asks the administrator to enter a path to the Tomcat folder in order to deploy the Library.war file and installs the WAR file in the `<INSTALL_PATH>/LibraryWebMobile` directory. Starting in 2019, administrators are no longer prompted with the Tomcat directory and can deploy the WAR file to the JEE server of choice.

Deploying the New WAR File

A new Library WAR File must be manually deployed after upgrading MicroStrategy Library, even between service packs.

You should backup the `webapps` folder as they are overwritten when deploying new `.war` files.

Use the following steps to deploy a new WAR file for both Windows and Linux.
1. Use the MicroStrategy installer to create a new MicroStrategy 2019 Library.war file by doing a direct upgrade or new installation.

2. Redeploy the new Library.war file from the Library install directory to the J2EE server being used. See the list of supported servers.

3. Migrate your customizations to the new environment, such as:
   - .../webapps/MicroStrategyLibrary/images/
   - .../webapps/MicroStrategyLibrary/plugins/
   - .../webapps/MicroStrategyLibrary/WEB-INF/classes/auth/
     - Customized authentication configuration, such as SAML authentication
   - .../webapps/MicroStrategyLibrary/WEB-INF/classes/config/configOverride.properties
     - Make sure MicroStrategy Library is pointing to the correct Intelligence Server and Collaboration Server with correct authentication setting by going to the Library Administration Control Panel.
   - .../webapps/MicroStrategyLibrary/WEB-INF/classes/logback.xml
     - Any other customizations including SSL configurations

4. Verify that MicroStrategy Library functionality including collaboration, sharing, exporting, printing, and all other features are working properly. Ensure dossiers and documents are appearing.

   The Collaboration Server is a separate component of MicroStrategy that is upgraded independently of Library. See step 9. MicroStrategy Collaboration for more information.
9. MicroStrategy Collaboration

MicroStrategy Collaboration has been enhanced to use a central data repository installed as part of the MicroStrategy Platform. Starting in MicroStrategy 2020 the Collaboration Server constructs the connection string to the MicroStrategy Data Repository during runtime. After performing the upgrade, Collaboration Server will only support connections to the MicroStrategy Data Repository.

After upgrading Collaboration Server, the previous data repository remains unchanged but it will not be used by the new version of Collaboration Server. Any comments stored in the previous data repository will need to be moved to the new repository to be accessible.

Migrating Comments to the MicroStrategy Data Repository

The Collaboration Server Administration Tool provides a way for system administrators to migrate comments from the previous data repository to the new data repository. The Collaboration Administration Page will display a warning message when it detects that a previous data repository was used by the Collaboration Server before the upgrade.

The Collaboration Server Administration Tool is a console application and will connect to the data repository using the same configuration as the Collaboration Server so no user input is required to connect. The tool will provide feedback as data is being migrated and will show a message when the migration is complete.

In order to recover collaboration messages that were stored in the previous version of the MicroStrategy Platform please follow these steps:

1. Stop the Collaboration service:

   ```
   cd
   ```
2. Start the previous data repository:

```
./collaborationServer.sh stop
```

3. Run the Data Migration Command:

   - Enter the three file locations in the following format:

     ```
     "$ "<PATH_TO_NODE>" "<PATH_TO_ADMIN_TOOL_FILE>" "<PATH_TO_CONFIG_FILE>"
     ```

     ```
     /opt/mstr/MicroStrategy/install/NodeJS/bin/node
     /opt/mstr/MicroStrategy/install/CollaborationServer/node_modules/mstr-collab-svc/admintool.js
     /opt/mstr/MicroStrategy/install/CollaborationServer/config.json
     ```

   - Use the `migrate` command to run the data migration:

     ```
     cmd> migrate
     ```

   - Enter `quit` to close the Collaboration Server Administration Tool.

4. Stop the service Mongo DB.


The Collaboration Admin Page will stop displaying the warning message at this point.
10. Updating the Enterprise Manager Project

Upgrading from MicroStrategy 9 to 2020 includes major enhancements and updates to the Enterprise Manager project; whereas upgrading from MicroStrategy 10 to 2020 does not have major enhancements and upgrades to the Enterprise Manager project. In both scenarios, you should use the latest versions of Enterprise Manager projects with MicroStrategy 2020.

To retain customizations created in previous Enterprise Manager projects, make separate copies of these customized copies of the out-of-the-box objects, or as brand-new objects. Any modifications you made to the out-of-the-box objects for an Enterprise Manager project are replaced as part of the upgrade process.

- To upgrade your Enterprise Manager statistics and warehouse, ensure your Enterprise Manager Data Loader service is pointing to your Statistics and Enterprise Manager Repository. For more information, see KB483298.

- Use the Project Duplication Wizard to make a backup of your existing Enterprise Manager project. For detailed information about using Project Duplication, see the Managing Your Projects chapter of the System Administration Guide.

1. Open Configuration Wizard.

2. Select Upgrade existing environment to MicroStrategy Secure Enterprise, and click Next.

3. Select Upgrade Enterprise Manager Project, and click Next.

4. Provide the following information:

   - **User Name**: Type the MicroStrategy user name that can access and administer the Enterprise Manager project.

   - **Password**: Type the password for the MicroStrategy user that can access and administer the Enterprise Manager project.
5. If you have an Enterprise Manager project package file (.mmp) to use instead of the default file, click Advanced >>. For the Package location field, click ... (the Browse button) to navigate to and select the package file.

6. Click Next.

7. Provide the following information:

   - **DSN**: Select the data source name for your statistics repository or 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.

8. Click Next.

9. Review the summary information.

   You can click Save to save the configuration as a response (.ini) file to upgrade your Enterprise Manager projects on other systems or to run silent configurations at a later time. For information on running the Configuration Wizard with a response file, see Using a Response File with Configuration Wizard.

10. Click Finish.

11. Upgrade the Platform Analytics Project

    As of 2019, you can upgrade your Platform Analytics project in the metadata of your connected Intelligence Server. Upgrading the project is recommended with each platform and update release in order to brings in the latest dossiers, attributes, metrics and reporting optimizations to the Platform Analytics project.
1. Open Configuration Wizard.

2. Select **Upgrade existing environment to MicroStrategy Secure Enterprise**, and click **Next**.

3. Select **Upgrade Platform Analytics Project**, and click **Next**.

4. Provide the following information:
   - **User Name**: Enter the MicroStrategy user name that can access the Intelligence Server.
     
     If this is your first time connecting to the MicroStrategy Intelligence Server, use the user name **Administrator** without a password.
   
   - **Password**: Enter the password for the MicroStrategy user that can access the Intelligence Server.

5. Choose the Platform Analytics Repository.

   Select the MySQL/PostgreSQL DSN for the Platform Analytics Repository that was used to create Platform Analytics.

6. Enter your **User Name** and **Password** for the DSN.

7. Click **Next**.

8. Click **Apply**. The Configuration Wizard automatically applies the following configuration files:
   - PlatformAnalyticsConfigurationNew.scp
   - PlatformAnalyticsConfigurationNew_PostgreSQL.scp
   - PlatformAnalyticsConfigurationUpgrade.scp
   - PlatformAnalyticsConfigurationUpgrade_PostgreSQL.scp
9. If an error appears about being unable to automatically apply project settings to Platform Analytics, you must manually update the project settings. For instructions, see Configure the Platform Analytics Project.

12. Upgrade the Platform Analytics Repository

After installing Platform Analytics, there is a Platform Analytics Repository (Platform Analytics warehouse table) where you can create the Platform Analytics project on your local machine. Upgrading the repository is required with each platform and update release in order to benefit from Platform Analytics warehouse new features, defect fixes and database structure optimizations.

The Configuration Wizard provides the following options:

- **Host**: Type the host name of the Platform Analytics warehouse. By default, this is set to the last successful connection value.

- **Port**: Type the port number of the Platform Analytics warehouse. By default, this is set to the last successful connection value.

- **User Name**: Type the user name for the Platform Analytics warehouse. By default, this is set to the value from PAConsumerConfig.yaml file

- **Password**: Type the password for the Platform Analytics warehouse user.

   Depending on the warehouse type you choose for the Host and Port, you must set the parameter `whDbType` to either "postgresql" or "mysql" in the PAConsumerConfig.yaml file.

   The default path is:

   - **Linux**: `/opt/MicroStrategy/PlatformAnalytics/Conf`
   - **Windows**: `C:\Program Files (x86)\MicroStrategy\Platform Analytics\conf`

   Click **Next** to proceed.
You can also update the Platform Analytics repository using the Configuration Wizard in interactive mode.

How to Update the Repository in Interactive Mode

To update the Platform Analytics repository using the Configuration Wizard in interactive mode on Windows:

1. In a Windows console, enter one of the following commands:
   - For 64-bit, enter `MACfgWiz_64`.
   - For 32-bit, enter `MACfgWiz`.
2. Click Enter.
3. Type 2 and click Enter to create a new `response.ini` file.
4. Type 5 and click Enter to upgrade your existing environment to MicroStrategy Analytics Enterprise.
5. Type 3 and click Enter to upgrade your Platform Analytics repository.
6. Enter your Platform Analytics warehouse database credentials. By default, the server name, port number, and user name are set to the last successful connection value.
   
   If you did not change the values, leave as default. The default password can be found at `C:\Program Files (x86)\Common Files\MicroStrategy\express_password.txt`
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 click **Enter**. The response.ini file is generated, and you are prompted whether to run the configuration immediately.

8. Type `Y` and click **Enter** to run the configuration.

To update the Platform Analytics repository using the Configuration Wizard in interactive mode on Linux:

1. In a Linux console window, browse to `HOME_PATH` where `HOME_PATH` is the specified home directory during installation.

2. Browse to the `bin` directory.

3. At the command prompt, type `mstrcfgwiz-editor`, then click **Enter**. The Configuration Wizard opens in command line mode.

4. Click **Enter**.

5. Type `2` and click **Enter** to create a new `response.ini` file.

6. Type `5` and click **Enter** to upgrade your existing environment to MicroStrategy Analytics Enterprise.

7. Type `3` and click Enter to upgrade your Platform Analytics repository.

8. Enter your Platform Analytics warehouse database credentials. By default, the server name, port number, and user name are set to the last successful connection value.

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 click **Enter**. The response.ini file is generated, and you
are prompted whether to run the configuration immediately.

10. Type `y` and click `Enter` to run the configuration.

13. Activate Your Installation

All MicroStrategy 2020 installations must be activated within 30 days. Once you receive the activation code you will need to enter it into MicroStrategy License Manager.

Instructions

1. Launch MicroStrategy License Manager:

   - **Command line**: In a Linux console window, browse to `/opt/mstr/MicroStrategy/bin` and run:
     ```bash
     ./mstrlicmgr -console
     ```

   - **GUI**: In a Linux console window, browse to `/opt/mstr/MicroStrategy/bin` and run:
     ```bash
     ./mstrlicmgr
     ```

2. Update your license key:

   - **Command line**:
     
     1. Choose option **2. Update local License key**.

     2. Enter your MicroStrategy 2020 license key when prompted and press `Enter`.

   - **GUI**

     1. Open the **License Administration** tab > **Activate Installation** and click Next.
2. Select **Server Activation Using Code** and enter your activation code in the text box. Click **Next**.

3. Click **OK**.

**Performing a Parallel Upgrade**

**Configuring an Upgrade Test Environment**

Your MicroStrategy environment includes multiple variables, such as security requirements, performance requirements, and VLDB settings, that are unique. MicroStrategy cannot anticipate all the ways these variables may interact with the upgrade process. Thus, MicroStrategy recommends you create a test environment and upgrade that environment first, then thoroughly test the upgraded installation. Once the tests are complete, then upgrade your production environment. This ensures that the upgrade of your production environment proceeds smoothly and any unexpected difficulties do not require additional downtime.

For detailed information on testing your upgraded environment, see *Chapter 4, Validating the Upgrade*.

If you do not want to create a test environment, MicroStrategy recommends that you create and save an Integrity Manager integrity test baseline of your reports and documents. You can then execute an integrity test against this baseline when the upgrade is complete, to ensure that the upgrade has not altered any of your report results. For detailed information about using Integrity Manager to execute integrity tests, see the *Integrity Manager* chapter of the *System Administration Guide*.

**Best Practices for Configuring an Upgrade Test Environment**

MicroStrategy recommends that you follow these best practices for configuring your upgrade test environment:
Do not modify any existing configuration objects. If you need additional configuration objects for testing, you can either create additional objects, or duplicate an existing object and modify it. This applies to database instances, connections and logins, security filters, users and user groups, and security roles.

If your production environment is clustered, then your test environment should also be clustered.

If your test and production data warehouses have different database table prefixes, make sure you are using the correct prefixes in the test environment's Warehouse Catalog.

Create an integrity test comparing reports from the upgraded test environment with the same reports in the production environment, so that you can easily see where any differences are.

If possible, plan to execute data integrity and performance load tests against the production warehouse. This ensures that the test scenarios are as representative of the production environment as possible.

If you are creating reports and documents specifically for an upgrade integrity test, create those reports and documents before you duplicate the production metadata.

If you are using connection mapping for users to access the data warehouse, check to be sure that all users can log in to the test data warehouse, since user passwords may differ between the test warehouse and the production warehouse.

One way to manage this is to create a new generic database login, and then use the following sample Command Manager script to change users' connection mappings to use this new login:

```
ALTER CONNECTION MAP FOR USER "username" DBINSTANCE "production_warehouse_instance" DBLOGIN "test_login" ON PROJECT "project";
```
If you are planning to upgrade Enterprise Manager, run a data load before you upgrade Intelligence Server. If you do not run a data load before upgrading Enterprise Manager and the projects on the Intelligence Servers, you may lose access to some statistics data. All Enterprise Manager upgrades require that you also upgrade your statistic tables, Enterprise Manager repository, and your Enterprise Manager metadata.

If you are using Narrowcast Server, make sure that the database copy of the Narrowcast repositories is not used when setting up the Narrowcast Server test environment. Instead, make a copy of the repositories with the Copy Repository utility included with Narrowcast Administrator and use this copy. This ensures that the test environment does not accidentally refer to a production server. For detailed instructions on creating a copy of the Narrowcast repositories, see the Narrowcast Server Upgrade Guide.

High-Level Steps to Configure an Upgrade Test Environment

To ensure that your tests accurately reflect the upgrade experience, the upgrade test environment should reflect the production environment as closely as possible.

To Configure a Test Environment

1. Set up the hardware for the environment. MicroStrategy recommends that this hardware duplicate the configuration of the production environment as closely as possible.

2. Install your current version of MicroStrategy in the test environment.

3. Using the Project Duplication Wizard, duplicate the production metadata into the test environment. For instructions on using the Project Duplication Wizard, see the Managing Your Projects chapter of the System Administration Guide, or see the Project Duplication Wizard Help.
4. Make sure that your test environment Intelligence Server is connected to your test environment metadata, and not your production metadata.

5. Using the Project Duplication Wizard, duplicate the production metadata into the test environment.

   For instructions on using the Project Duplication Wizard, see the Managing Your Projects chapter of the System Administration Guide, or see the Project Duplication Wizard Help.

6. Ensure your test environment Intelligence Server is connected to your test environment metadata.

7. If you do not intend to execute your tests against a production warehouse, duplicate the production warehouse, and ensure that the test environment points to the duplicate warehouse and not the production warehouse.

8. Upgrade the test environment, following the procedures laid out in Performing a Parallel Upgrade on a Windows Deployment, page 71

9. Test the upgrade, following the guidelines laid out in Validating the Upgrade, page 155.

Performing a Parallel Upgrade on a Windows Deployment

This section covers the procedure for executing a parallel upgrade of your Windows environment to MicroStrategy 2020. Carefully review the Planning Your Upgrade and Configuring an Upgrade Test Environment sections before proceeding with your upgrade.

1. Install MicroStrategy 2020

Follow the procedures outlined in the Installation and Configuration Guide to perform a fresh installation of MicroStrategy 2020 on your test environment.
Once you have installed and configured your test environment, proceed with Step 2. **Connect the Intelligence Server to a Copy of Your Metadata.**

2. Connect the Intelligence Server to a Copy of Your Metadata

Before connecting your new Intelligence Server to the metadata, make a copy of your production metadata in your database. You will connect the Intelligence Server to this metadata copy in the following steps.

> Repeat these steps for each Intelligence server node in a clustered environment.

Configure Metadata Connection

1. In Windows, go to **Start > All Programs > MicroStrategy Tools > Configuration Wizard**. The Configuration Wizard opens.

2. Select **Configure Intelligence Server**, and click **Next**.

3. From the **DSN** drop-down list, select the DSN used to connect to the metadata repository. Enter the **User Name** and **Password** for the database.

4. Click **Next**.

5. Select **Use the selected Server Definition as active.**

6. Choose the server definition to use from the **Existing Server Definitions** pane.

7. Click **Next**.

Define the Intelligence Server port number and other settings

1. Define the Intelligence Server settings described below:

   * **Port number**: You can use the default port number (34952) or specify another port number. The port number is how a server
process identifies itself on the machine on which it is running. If the port number is used by another process, such as in a shared environment, specify an available port number.

- **REST port number**: You can use the default port number (34962) or specify another port number for the REST API Server inside Intelligence Server. This port number should not be same with the Intelligence Server port number.

  You can configure REST API Server logging with the Diagnostics and Performance Logging Tool. Select the **Performance Configuration** tab, and find the dispatcher **REST Trace** under component **Network Classes**. For more information, see the Configuring What is Logged section in the **System Administration Guide**.

- **Register Intelligence Server as a Service**: Select this check box to register Intelligence Server as a service.

- **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.

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

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

2. Click Next.

3. Secure socket layer (SSL) protocol encrypts communications between Intelligence Server and Developer:

   **Configure SSL**: This option specifies whether to enable Intelligence Server and Developer to communicate using the SSL protocol. Clear this check box to disable the use of the SSL protocol for Intelligence Server and Developer communications. This option also applies SSL protocol to the REST port number.

   When you select to enable the SSL protocol, you must provide the following information:

   - **Certificate**: The SSL certificate file you created for Intelligence Server. Click the browse button to navigate to and select the certificate file.

   - **Key**: The private key file you created while requesting the certificate for Intelligence Server. Click the browse button to navigate to and select the private key file.

   - **Password**: The password that you used while creating the private key for the SSL certificate.

   - **SSL Port**: The port number to use for SSL access. By default, the port is 39321.

   To enable SSL protocol communication in Developer, you must use the Project Source Editor. For steps to complete the other tasks required to enable SSL protocol communications, refer to the System Administration Guide.
Configuring port requires Client Certificate: If selected, the SSL communications with client certificate verification will be configured in the Intelligence Server, but does not apply to REST port number. Provide the following information:

- **SSL Port**: The port number the Intelligence Server will use for SSL communications with client certificate verification.
- **Truststore**: The location to the client certificate truststore.

4. Click **Next**.

5. To specify the default statistics repository for the local Intelligence Server and 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)**: Basic statistics for new
projects and any projects that are not currently logging statistics will be enabled. You can alter the settings in Project Configuration Editor in MicroStrategy Developer to:

- Enable additional statistics for a project.
- Enable basic and additional statistics for a project if you cleared this check box.

6. Enable and configure the Messaging Services host and port settings.

**Messaging Services Host(s)/Port(s):** Provide host name or IP address of Messaging Services.

*Format for this setting should be:* server1.acme.com:9092

7. Click **Next**.

8. Review the summary information and click **Save**.

9. Click **Finish**.

3. Update the Metadata Copy

After upgrading the Intelligence Server, update the copy of your metadata by performing the procedure that follows. This procedure creates and updates metadata tables to support new features available in MicroStrategy 2020.

Updating the metadata ensure compatibility between your pre-existing objects and the MicroStrategy clients. You can also take advantage of new configuration settings exposed through the client applications, functionality, and objects available only with the updated version of metadata.

During the metadata update process, the Configuration Wizard generates additional SQL for tasks such as managing indexes and primary keys. To view this SQL during the update process, on the Summary Page of the Configuration Wizard, in the Summary pane, click **SQL Preview**. In addition, once you have completed the update process, the generated SQL
is saved in the MicroStrategy common files directory. The file name is `OriginalScript_generated.sql`, where `OriginalScript` is the name of the original SQL update script. For example, if you are updating an Oracle database, the original SQL update script is `mdorcl.sql` and the generated SQL file is `mdorcl_generated.sql`.

Be aware of the following:

- If you have made any changes to the privileges assigned to the out-of-the-box MicroStrategy user groups and security roles, updating the metadata may overwrite those changes.

- For information about viewing the Configuration Wizard log file if an error occurs, see *Resolving Problems Encountered During the Upgrade*.

- The MDUpdate command line utility that was formerly used to update the metadata is no longer supported. To update the metadata from the command line, you can use a response file with Configuration Wizard.

As an alternative to stepping through each page of the Configuration Wizard for each project source that needs to be updated, you can create a response file with the update information and use that response file with the Configuration Wizard to automatically update your metadata. For more information about using a response file to update the metadata, see *Using a Response File with Configuration Wizard*.

If you do not have access to the GUI mode of Configuration Wizard, you must use a response file to update your metadata.

**Prerequisites for Updating a MicroStrategy Metadata**

Before attempting to update your MicroStrategy metadata to the latest version, make sure you meet the following prerequisites:

- The metadata update process can be executed only by a MicroStrategy user who is either a member of the System Administrators user group, or
is the out-of-the-box Administrator user. Having all administrative privileges is not sufficient: the user must be a member of the System Administrators user group.

- The project's metadata version from which you update must be older than, or the same as, the version of the machine where Configuration Wizard is installed that will be performing the metadata update. An older version of the product is not able to perform an update on a newer version of metadata.

- The language settings of the client, project, and Intelligence Server must all be the same.

- The projects to be updated must not be locked. To unlock a project's metadata, in Developer, from the Administration menu, go to Locking > Unlock Project.

Database-Specific Prerequisites

The following prerequisites are specific to the RDBMS that your project metadata is stored in:

- If you are using Oracle for your metadata database, make sure that the Maximum Open Cursors parameter for the database is set to at least 1500.

- If you are using SQL Server for your metadata database, MicroStrategy recommends setting the transaction log to the Simple Recovery Model because the Bulk or Full Recovery Models may fill the transaction log during the upgrade and cause an error.

- If you are using DB2 for your metadata database, make sure that the Dynamic Sections property for the database is set to 999.
To Make Metadata Updates Using the Configuration Wizard

1. Select the **Upgrade existing environment to MicroStrategy Secure Enterprise** option. Click **Next**.

2. Select the **Intelligence Server components** option. Click **Next**.

3. On the MicroStrategy Authentication page, type the username and password of a MicroStrategy system administrator. Click **Next**.

4. On the Metadata Connection page, enter the database password to access the MicroStrategy metadata. Click **Next**.

   The DSN and Login ID are provided based on the server configuration that you did previously.

5. On the Select Components page, select the check boxes for each Intelligence Server you want to upgrade.

6. Under each selected Intelligence Server, select the system components you want to upgrade:

   - **Upgrade metadata repository**: The metadata repository contains the definitions of your MicroStrategy applications and supporting objects. An upgrade of your metadata is required to provide support for all new and updated features in the most recent version of MicroStrategy.

   - **Lean Objects migration**: Lean Objects are a significantly more compact form of object representation in the metadata tables. Lean Objects are typically 25-50% smaller and will never be larger than the original representation. Migrating existing objects to Lean Objects also allows for faster loading of objects compared to the old representation.

   All new objects are created in this format, and existing objects are converted to the new format as they are saved. Selecting Lean Objects migration converts all objects in the repository to the new format.
format at once. This option is automatically selected if you select Upgrade metadata repository.

The Lean objects migration modifies all objects in your MicroStrategy projects to use the new object representation. This migration can require a significant amount of time and resources to complete.

Lean Objects migration decreases the amount of disk space used by your objects. However, the migration causes all objects in your projects to be resaved. Depending on your database transaction logging policy, this may cause a temporary increase in the size of your metadata database.

- **Update privileges**: Updating the privileges ensures that users have access to the same functionality that they had in previous versions of MicroStrategy.

  By default, privileges are upgraded based on the version of your MicroStrategy metadata. However, if you previously upgraded your MicroStrategy metadata and did not upgrade privileges along with the metadata upgrade, you must identify the MicroStrategy version you most recently upgraded your privileges for.

  To do this, to the right of **Update privileges**, click **Advanced**. The Select Privilege Version dialog box opens, with the version of your MicroStrategy metadata selected by default. From the drop-down list, select the version of MicroStrategy that privileges were most recently upgraded for and click **OK**.

- **Migrate History List messages**: Select this check box to migrate your existing History List repository to a new format for improved search results, scalability, and performance. A file-based repository can be migrated to a database-based repository or a hybrid repository, and a database-based repository can be migrated to a
hybrid repository.

- If you select this check box, additional configuration options for this migration are provided later, on the History List Migration page. During the upgrade process this Intelligence Server and any other Intelligence Servers in the cluster are stopped and restarted.

- If you are using a file-based History List repository and want to continue using that format, clear this check box. The repository is updated, but its format is not affected, when you restart the Intelligence Server after the upgrade.

- If you are using a database-based or hybrid History List repository and want to continue using the same type of repository, for information about upgrading the History List database, see 4. History List Database Repository.

- **Update Distribution Services Objects**: Select this check box to upgrade Distribution Services subscription objects. This update is required to ensure that your Distribution Services subscriptions can support the new features and enhancements included in the most recent release. If you select this option, ensure that the metadata repository has been upgraded or is selected to be upgraded as part of this update.

- **Database Instance Update**: Select this checkbox to update any MDX database instances which currently use JCO 2.x to JCO 3.x. As of MicroStrategy 10.x the Java Virtual Machine used does not support JCO 2.x. This option will only affect MDX database instances which are set to use JCO 2.x.

7. Select each project you want to update. Only projects that are loaded on Intelligence Server can be updated. Project updates can include any of the following options:
• **Execute project logical upgrade**: Updates the project to the most recent version of MicroStrategy. This option also updates the report, document, and dossier definitions to improve the performance and reduce the memory usage and storage requirements of reports and documents that were created with a previous version of MicroStrategy.

Be aware of the following:

• If this update has already been completed, this option is named **Re-execute project logical upgrade**.

• This upgrade is required to execute any additional project upgrade options.

• Updated Report Services documents cannot be run in previous versions of MicroStrategy.

• Depending on the number of reports and documents in the project, this update may take significant time.

• **Lean Objects migration**: Selecting Lean Objects migration converts all objects in this project to the new format at once.

The Lean objects migration modifies all objects in this project to use the new object representation. This migration can require significant time and resources to complete.

Lean Objects migration decreases the amount of disk space used by your objects. However, the migration causes all objects in this project to be resaved. Depending on your database transaction logging policy, this may cause a temporary increase in the size of your metadata database.

• **Update Schedules**: Makes previous subscriptions viewable in MicroStrategy 2020.
- **Update MDX Source Objects:** Updates the MDX data source objects (MDX Cubes) that were created in earlier versions of MicroStrategy to take advantage of improved performance.

  By default, if you update multiple projects and the update fails for one project, the update process continues for other projects. This allows you to complete all possible updates and then review any errors. To abort the update process for other projects when one project fails, clear the **Update the remaining projects, even when one project update fails** check box.

  Depending on the number of MDX source objects in the project, this update may take significant time.

- **Update OLAP Services Cube caches to PRIME:** Select this check box to update all Intelligent Cubes to the PRIME architecture. This architecture can improve the performance of publishing Intelligent Cube results through the use of parallel processing and other optimizations. This update is applied to all Intelligent Cubes for the project.

- **Update data import cubes (definition and cache) to PRIME:** Select this check box to update all Data Import datasets to the PRIME architecture. This architecture can improve the performance of publishing Data Import results through the use of parallel processing and other optimizations. This update is applied to data sets included in your project through the use of Data Import.

  It is easier to update OLAP Services Cube caches and data import cubes to PRIME during the upgrade process. If you do not update them now, each cube is updated when it is used. Doing the upgrade all at once ensures consistent performance for end users.

8. Click **Next**.
9. If you selected the Migrate History List messages option, on the History List Target Database page, specify the necessary information for the History List database.

10. Click Next.

11. On the Summary page, review your upgrade choices and click Finish.

4. History List Database Repository

If you are using a database-based History List, when upgrading to MicroStrategy 2020 you must create the History List database tables in the test environment.

To Create the Tables in the History List Repository

1. Open the MicroStrategy Configuration Wizard.

2. Select Metadata, History List and Statistics Repository Tables and click Next.

3. Select the History List Tables check box and click Next.

4. From the DSN drop-down list, select the DSN for your History List repository.

   If a DSN for your History List repository does not exist, you can select New to open the Connectivity Wizard and create a new DSN.

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

   The database user you provide becomes the owner of all History List tables and objects. The database user is required to have the Select, Create, Insert, and Drop permissions.

6. After providing a valid user name and password, you can click SQL Preview to open the SQL Preview dialog box. This dialog box provides the SQL statements that will be executed on your data source to create the History List tables. Click Close once you are done reviewing the
SQL statements to return to the Configuration Wizard.

If you use the advanced options to change the SQL script, you can click SQL Preview after selecting the new script to see an updated listing of the SQL statements that will be executed.

7. Click **Advanced**.

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

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

If you use a table prefix for your History List tables, you must also define this table prefix when you create a database instance to connect to the History List tables.

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

10. Click **Next**.

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.

12. Click **Finish** to apply the configuration and create the History List repository.
5.1 Updating the Enterprise Manager Project

Upgrading from MicroStrategy 9 to 2020 includes major enhancements and updates to the Enterprise Manager project; whereas upgrading from MicroStrategy 10 to 2020 does not have major enhancements and upgrades to the Enterprise Manager project. In both scenarios, you should use the latest versions of Enterprise Manager projects with MicroStrategy 2020.

To retain customizations created in previous Enterprise Manager projects, make separate copies of these customized copies of the out-of-the-box objects, or as brand-new objects. Any modifications you made to the out-of-the-box objects for an Enterprise Manager project are replaced as part of the upgrade process.

- To upgrade your Enterprise Manager statistics and warehouse, ensure your Enterprise Manager Data Loader service is pointing to your Statistics and Enterprise Manager Repository. For more information, see KB483298.

- Use the Project Duplication Wizard to make a backup of your existing Enterprise Manager project. For detailed information about using Project Duplication, see the Managing Your Projects chapter of the System Administration Guide.

1. Open Configuration Wizard.
2. Select Upgrade existing environment to MicroStrategy Secure Enterprise, and click Next.
3. Select Upgrade Enterprise Manager Project, and click Next.
4. Provide the following information:
   - **User Name**: Type the MicroStrategy user name that can access and administer the Enterprise Manager project.
   - **Password**: Type the password for the MicroStrategy user that can access and administer the Enterprise Manager project.
5. If you have an Enterprise Manager project package file (.mmp) to use instead of the default file, click Advanced >>. For the Package location field, click ... (the Browse button) to navigate to and select the package file.

6. Click Next.

7. Provide the following information:

   - **DSN**: Select the data source name for your statistics repository or 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.

8. Click Next.

9. Review the summary information.

   You can click Save to save the configuration as a response (.ini) file to upgrade your Enterprise Manager projects on other systems or to run silent configurations at a later time. For information on running the Configuration Wizard with a response file, see Using a Response File with Configuration Wizard.

10. Click Finish.

6. Upgrade the Platform Analytics Project

As of 2019, you can upgrade your Platform Analytics project in the metadata of your connected Intelligence Server. Upgrading the project is recommended with each platform and update release in order to brings in the latest dossiers, attributes, metrics and reporting optimizations to the Platform Analytics project.
1. Open Configuration Wizard.
2. Select **Upgrade existing environment to MicroStrategy Secure Enterprise**, and click **Next**.
3. Select **Upgrade Platform Analytics Project**, and click **Next**.
4. Provide the following information:
   - **User Name**: Enter the MicroStrategy user name that can access the Intelligence Server.
     
     If this is your first time connecting to the MicroStrategy Intelligence Server, use the user name **Administrator** without a password.
   - **Password**: Enter the password for the MicroStrategy user that can access the Intelligence Server.
5. Choose the Platform Analytics Repository.
   
   Select the MySQL/PostgreSQL DSN for the Platform Analytics Repository that was used to create Platform Analytics.
6. Enter your **User Name** and **Password** for the DSN.
7. Click **Next**.
8. Click **Apply**. The Configuration Wizard automatically applies the following configuration files:
   - PlatformAnalyticsConfigurationNew.scp
   - PlatformAnalyticsConfigurationNew_PostgreSQL.scp
   - PlatformAnalyticsConfigurationUpgrade.scp
   - PlatformAnalyticsConfigurationUpgrade_PostgreSQL.scp
9. If an error appears about being unable to automatically apply project settings to Platform Analytics, you must manually update the project settings. For instructions, see Configure the Platform Analytics Project.

7. Migrate Platform Analytics Data to MicroStrategy Repository

The Platform Analytics Data Migration tool is used to help existing customers migrate their data from MySQL to the newly supported PostgreSQL repository. This tool can help migrate both new and old versions of MySQL dump files to the latest version of Platform Analytics.

**Backup Prerequisites:**

- C:\Program Files (x86)\MicroStrategy\Platform Analytics\PAConsumerConfig.yaml populated with:

  warehouseDbConnection:
  
  - whHost: 127.0.0.1
  - whUser: root
  - whPasswd: encrypted_password
  - whPort: 3306
  - whDb: platform_analytics_wh

- mysql-connector-java.jar is present in PlatformAnalytics\lib directory.

- Disk space sufficient to hold a backup of your MySQL platform_analytics_wh database.

**Restore Prerequisites:**

- PAConsumerConfig.yaml populated with:

  pgWarehouseDbConnection:
Launching the Platform Analytics Data Migration Tool

1. Navigate to your Platform Analytics home directory and go into the bin directory:

   C:\Program Files (x86)\MicroStrategy\Platform Analytics\bin

2. Call the following script:

   platform-analytics-data-migration-tool.bat

3. You will then be prompted with the following:

   This is the Platform Analytics Data Migration Tool. The purpose of this tool is to help migrate your data from an existing Mysql Warehouse to a new PostgreSQL Warehouse.

   Please select from the following:
   1) Backup
   2) Restore
   3) Backup and Restore
   0) Exit
Migration Workflow

Backup

1. Provide the path to the directory where the MySQL backup will be stored.

2. The tool will then begin backing up the MySQL platform-analytics_wh specified in your PAConsumerConfig.yaml file, placing the backup in your specified path.

Restore

1. Provide the path to the directory where the MySQL backup is stored.

2. The tool will prompt you again if you are sure you are okay to drop your PostgreSQL platform-analytics_wh schema.

3. If yes is selected, the platform-analytics_wh schema will be dropped and recreated matching the version of your MySQL dump.

4. The backup data is then imported into the newly created platform-analytics_wh schema.

5. The platform-analytics_wh schema will then be upgraded to the latest version of Platform Analytics.

Recommended Upgrade Procedures

1. On your new MicroStrategy 2020 machine, populate the PAConsumerConfig.yaml has the MySQL and PostgreSQL information shown in the prerequisites above.

2. Copy the mysql-connector-java.jar from your previous installation to the Platform Analytics\lib directory on the new machine.
3. Go to your Platform Analytics bin directory and call the `platform-analytics-data-migration-tool.bat` file.

4. Select the Backup and Restore option (3).

5. Enter the full desired directory path for the database to be backed up to and restored from.

6. Wait until the backup is complete. The tool you will then prompt if it is okay to recreate the PostgreSQL warehouse and select yes.

7. The program will then restore your MySQL backup files into your new PostgreSQL warehouse and the data migration will be complete.

8. Migrate Comments to a New Installation of Collaboration Server

In MicroStrategy 2020, Collaboration Server uses MicroStrategy Repository to store user comments. When performing a parallel upgrade, the new installation of Collaboration Server will be automatically configured to MicroStrategy Repository during installation. After performing the upgrade, Collaboration Server will only support connections to the MicroStrategy Data Repository. Any existing user comments will need to be migrated to the new repository to remain accessible.

The Collaboration Server Administration Tool provides a way for system administrators to migrate comments from the previous data repository to the new data repository. The Collaboration Administration Page will display a warning message when it detects that a previous data repository was used by the Collaboration Server before the upgrade.

The Collaboration Server Administration Tool is a console application and will connect to the data repository using the same configuration as the Collaboration Server so no user input is required to connect. The tool will provide feedback as data is being migrated and will show a message when the migration is complete.
On the existing Collaboration Server:

1. **Edit the mongod.cfg file under the C:\Program Files\MicroStrategy\Collaboration Server\MongoDB directory**, add a new section `net` with `bindIpAll: true`. It will allow the MongoDB to be accessible outside of the machine so we can use the Collaboration Server Administration Tool to connect to the MongoDB from another machine.

   Make sure to use four spaces instead of a tab for the indentation because the `mongod.cfg` is in YAML format.

   ```yaml
   ... 
   processManagement:
     windowsService:
       serviceName: MSTR_mongodb
       displayName: MongoDB
     security:
       authorization: enabled
     net:
       bindIpAll: true
   ...
   ```

2. **Click Start > Windows Administrative Tools > Services.**

3. **Restart the MongoDB service to apply the modified configuration.**

4. **Open the config.json file under the C:\Program Files\MicroStrategy\Collaboration Server directory, copy the whole line of the dburl property.**

On the new Collaboration Server machine:

1. **Click Start > Windows Administrative Tools > Services.**

2. **Stop the MicroStrategy Collaboration/Realtime Service.**

3. **Open the config.json file under the C:\Program...**
Files\MicroStrategy\Collaboration Server directory, paste the dburl line at the top of the content. Replace the localhost string in the dburl with the IP of the existing 2019 machine. Make sure the content is in valid JSON format. Here is an example of the modified config.json file.

```json
{
   "dburl": "mongodb://mstr:xxxx@[IP of the existing 2019 machine]:27017/mstr_collab?authSource=admin",
   "port": 3000,
   "logging": false,
   "authorizationServerUrl": "http://localhost:8080/MicroStrategyLibrary/api",
   "dataSource": {
      "username": "mstr_collab",
      "password": "xxxxxx"
   },
   "scaling": "none",
   "secretKey": "xxxx",
   "enableConfigApi": true
}
```

4. Run the Data Migration Command:

- Enter the three file locations in the following format:

```
$ "<PATH_TO_NODE>" "<PATH_TO_ADMIN_TOOL_FILE>" "<PATH_TO_CONFIG_FILE>"
```

- Use the migrate command to run the data migration:

8. Activate Your Installation

All MicroStrategy 2020 installations must be activated within 30 days. Once you receive the activation code you will need to enter it into MicroStrategy License Manager.

Instructions

1. Open MicroStrategy License Manager from Start menu > Programs > MicroStrategy Tools > License Manager.

2. Go to the License Administration tab > Activate Server Installation and click Next.

3. Select the Server Activation using Activation Code option and enter your Activation Code in the text field. Click Next.

4. Click OK.

Performing a Parallel Upgrade on a Linux Deployment

This section covers the procedure for executing a parallel upgrade of your Linux environment to MicroStrategy 2020. Carefully review the Planning Your Upgrade and Configuring an Upgrade Test Environment sections before proceeding with your upgrade.

1. Install MicroStrategy 2020

Follow the procedures outlined in the Installation and Configuration Guide to perform a fresh installation of MicroStrategy 2020 on your test environment. Once you have installed and configured your test
2. Connect the Intelligence Server to a Copy of Your Metadata

Before connecting your new Intelligence Server to the metadata, make a copy of your production metadata in your database. You will connect the Intelligence Server to this metadata copy in the following steps.

Repeat these steps for each Intelligence server node in a clustered environment.

Configure Metadata Connection

1. Run the MicroStrategy Configuration Wizard:

   ```bash
   cd /opt/mstr/Microstrategy/bin
   ./mstrcfgwiz
   ```

2. Select **Configure Intelligence Server**, and click **Next**.

3. From the **DSN** drop-down list, select the DSN used to connect to the metadata repository. Enter the **User Name** and **Password** for the database.

4. Click **Next**.

5. Select **Use the selected Server Definition as active**.

6. Choose the server definition to use from the **Existing Server Definitions** pane.

7. Click **Next**.
Define the Intelligence Server port number and other settings

1. Define the Intelligence Server settings described below:

- **Port number**: You can use the default port number (34952) or specify another port number. The port number is how a server process identifies itself on the machine on which it is running. If the port number is used by another process, such as in a shared environment, specify an available port number.

- **REST port number**: You can use the default port number (34962) or specify another port number for the REST API Server inside Intelligence Server. This port number should not be the same with the Intelligence Server port number.

You can configure REST API Server logging with the Diagnostics and Performance Logging Tool. Select the Performance Configuration tab, and find the dispatcher **REST Trace** under component **Network Classes**. For more information, see the Configuring What is Logged section in the System Administration Guide.

- **Register Intelligence Server as a Service**: Select this check box to register Intelligence Server as a service.

- **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.

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

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

2. Click **Next**.

3. Secure socket layer (SSL) protocol encrypts communications between Intelligence Server and Developer:

   **Configure SSL**: This option specifies whether to enable Intelligence Server and Developer to communicate using the SSL protocol. Clear this check box to disable the use of the SSL protocol for Intelligence Server and Developer communications. This option also applies SSL protocol to the REST port number.

   When you select to enable the SSL protocol, you must provide the following information:

   - **Certificate**: The SSL certificate file you created for Intelligence Server. Click the browse button to navigate to and select the certificate file.
   
   - **Key**: The private key file you created while requesting the certificate for Intelligence Server. Click the browse button to navigate to and select the private key file.
   
   - **Password**: The password that you used while creating the private key for the SSL certificate.
   
   - **SSL Port**: The port number to use for SSL access. By default, the port is 39321.
To enable SSL protocol communication in Developer, you must use the Project Source Editor. For steps to complete the other tasks required to enable SSL protocol communications, refer to the System Administration Guide.

**Configuring port requires Client Certificate**: If selected, the SSL communications with client certificate verification will be configured in the Intelligence Server, but does not apply to REST port number. Provide the following information:

- **SSL Port**: The port number the Intelligence Server will use for SSL communications with client certificate verification.

- **Truststore**: The location to the client certificate truststore.

4. Click **Next**.

5. To specify the default statistics repository for the local Intelligence Server and 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)**: Basic statistics for new projects and any projects that are not currently logging statistics will be enabled. You can alter the settings in Project Configuration Editor in MicroStrategy Developer to:
  ᵉ ○ Enable additional statistics for a project.
  ᵉ ○ Enable basic and additional statistics for a project if you cleared this check box.

6. Enable and configure the Messaging Services host and port settings.

  **Messaging Services Host(s)/Port(s)**: Provide host name or IP address of Messaging Services.
  
  Format for this setting should be: server1.acme.com:9092

7. Click **Next**.

8. Review the summary information and click **Save**.

9. Click **Finish**.

3. Update the Metadata Copy

After upgrading the Intelligence Server, update the copy of your metadata by performing the procedure that follows. This procedure creates and updates metadata tables to support new features available in MicroStrategy 2020.

Updating the metadata ensure compatibility between your pre-existing objects and the MicroStrategy clients. You can also take advantage of new configuration settings exposed through the client applications, functionality, and objects available only with the updated version of metadata.
During the metadata update process, the Configuration Wizard generates additional SQL for tasks such as managing indexes and primary keys. To view this SQL during the update process, on the Summary Page of the Configuration Wizard, in the Summary pane, click **SQL Preview**. In addition, once you have completed the update process, the generated SQL is saved in the MicroStrategy common files directory. The file name is `OriginalScript_generated.sql`, where `OriginalScript` is the name of the original SQL update script. For example, if you are updating an Oracle database, the original SQL update script is `mdorcl.sql` and the generated SQL file is `mdorcl_generated.sql`.

Be aware of the following:

- If you have made any changes to the privileges assigned to the out-of-the-box MicroStrategy user groups and security roles, updating the metadata may overwrite those changes.

⚠️ For information about viewing the Configuration Wizard log file if an error occurs, see *Resolving Problems Encountered During the Upgrade*.

- The MDUpdate command line utility that was formerly used to update the metadata is no longer supported. To update the metadata from the command line, you can use a response file with Configuration Wizard.

As an alternative to stepping through each page of the Configuration Wizard for each project source that needs to be updated, you can create a response file with the update information and use that response file with the Configuration Wizard to automatically update your metadata. For more information about using a response file to update the metadata, see *Using a Response File with Configuration Wizard*.

⚠️ If you do not have access to the GUI mode of Configuration Wizard, you must use a response file to update your metadata.
Prerequisites for Updating a MicroStrategy Metadata

Before attempting to update your MicroStrategy metadata to the latest version, make sure you meet the following prerequisites:

- The metadata update process can be executed only by a MicroStrategy user who is either a member of the System Administrators user group, or is the out-of-the-box Administrator user. Having all administrative privileges is not sufficient: the user must be a member of the System Administrators user group.

- The project’s metadata version from which you update must be older than, or the same as, the version of the machine where Configuration Wizard is installed that will be performing the metadata update. An older version of the product is not able to perform an update on a newer version of metadata.

- The language settings of the client, project, and Intelligence Server must all be the same.

- The projects to be updated must not be locked. To unlock a project’s metadata, in Developer, from the Administration menu, go to Locking > Unlock Project.

Database-Specific Prerequisites

The following prerequisites are specific to the RDBMS that your project metadata is stored in:

- If you are using Oracle for your metadata database, make sure that the Maximum Open Cursors parameter for the database is set to at least 1500.

- If you are using SQL Server for your metadata database, MicroStrategy recommends setting the transaction log to the Simple Recovery Model because the Bulk or Full Recovery Models may fill the transaction log during the upgrade and cause an error.
If you are using DB2 for your metadata database, make sure that the Dynamic Sections property for the database is set to 999.

**To Make Metadata Updates Using the Configuration Wizard**

1. Select the **Upgrade existing environment to MicroStrategy Secure Enterprise** option. Click **Next**.

2. Select the **Intelligence Server components** option. Click **Next**.

3. On the MicroStrategy Authentication page, type the username and password of a MicroStrategy system administrator. Click **Next**.

4. On the Metadata Connection page, enter the database password to access the MicroStrategy metadata. Click **Next**.

   The DSN and Login ID are provided based on the server configuration that you did previously.

5. On the Select Components page, select the check boxes for each Intelligence Server you want to upgrade.

6. Under each selected Intelligence Server, select the system components you want to upgrade:

   - **Upgrade metadata repository**: The metadata repository contains the definitions of your MicroStrategy applications and supporting objects. An upgrade of your metadata is required to provide support for all new and updated features in the most recent version of MicroStrategy.

   - **Lean Objects migration**: Lean Objects are a significantly more compact form of object representation in the metadata tables. Lean Objects are typically 25-50% smaller and will never be larger than the original representation. Migrating existing objects to Lean Objects also allows for faster loading of objects compared to the old representation.
All new objects are created in this format, and existing objects are converted to the new format as they are saved. Selecting Lean Objects migration converts all objects in the repository to the new format at once. This option is automatically selected if you select Upgrade metadata repository.

The Lean objects migration modifies all objects in your MicroStrategy projects to use the new object representation. This migration can require a significant amount of time and resources to complete.

Lean Objects migration decreases the amount of disk space used by your objects. However, the migration causes all objects in your projects to be resaved. Depending on your database transaction logging policy, this may cause a temporary increase in the size of your metadata database.

- **Update privileges**: Updating the privileges ensures that users have access to the same functionality that they had in previous versions of MicroStrategy.

  By default, privileges are upgraded based on the version of your MicroStrategy metadata. However, if you previously upgraded your MicroStrategy metadata and did not upgrade privileges along with the metadata upgrade, you must identify the MicroStrategy version you most recently upgraded your privileges for.

  To do this, to the right of **Update privileges**, click **Advanced**. The Select Privilege Version dialog box opens, with the version of your MicroStrategy metadata selected by default. From the drop-down list, select the version of MicroStrategy that privileges were most recently upgraded for and click **OK**.

- **Migrate History List messages**: Select this check box to migrate your existing History List repository to a new format for improved
search results, scalability, and performance. A file-based repository can be migrated to a database-based repository or a hybrid repository, and a database-based repository can be migrated to a hybrid repository.

- If you select this check box, additional configuration options for this migration are provided later, on the History List Migration page. During the upgrade process this Intelligence Server and any other Intelligence Servers in the cluster are stopped and restarted.

- If you are using a file-based History List repository and want to continue using that format, clear this check box. The repository is updated, but its format is not affected, when you restart the Intelligence Server after the upgrade.

- If you are using a database-based or hybrid History List repository and want to continue using the same type of repository, for information about upgrading the History List database, see 4. History List Database Repository.

- **Update Distribution Services Objects**: Select this check box to upgrade Distribution Services subscription objects. This update is required to ensure that your Distribution Services subscriptions can support the new features and enhancements included in the most recent release. If you select this option, ensure that the metadata repository has been upgraded or is selected to be upgraded as part of this update.

- **Database Instance Update**: Select this checkbox to update any MDX database instances which currently use JCO 2.x to JCO 3.x. As of MicroStrategy 10.x the Java Virtual Machine used does not support JCO 2.x. This option will only affect MDX database instances which are set to use JCO 2.x.

7. Select each project you want to update. Only projects that are loaded on Intelligence Server can be updated. Project updates can include
any of the following options:

- **Execute project logical upgrade**: Updates the project to the most recent version of MicroStrategy. This option also updates the report, document, and dossier definitions to improve the performance and reduce the memory usage and storage requirements of reports and documents that were created with a previous version of MicroStrategy.

  Be aware of the following:

  - If this update has already been completed, this option is named **Re-execute project logical upgrade**.
  
  - This upgrade is required to execute any additional project upgrade options.
  
  - Updated Report Services documents cannot be run in previous versions of MicroStrategy.
  
  - Depending on the number of reports and documents in the project, this update may take significant time.

- **Lean Objects migration**: Selecting Lean Objects migration converts all objects in this project to the new format at once.

  The Lean objects migration modifies all objects in this project to use the new object representation. This migration can require significant time and resources to complete.

  Lean Objects migration decreases the amount of disk space used by your objects. However, the migration causes all objects in this project to be resaved. Depending on your database transaction logging policy, this may cause a temporary increase in the size of your metadata database.
- **Update Schedules**: Makes previous subscriptions viewable in MicroStrategy 2020.

- **Update MDX Source Objects**: Updates the MDX data source objects (MDX Cubes) that were created in earlier versions of MicroStrategy to take advantage of improved performance.

  By default, if you update multiple projects and the update fails for one project, the update process continues for other projects. This allows you to complete all possible updates and then review any errors. To abort the update process for other projects when one project fails, clear the **Update the remaining projects, even when one project update fails** check box.

  Depending on the number of MDX source objects in the project, this update may take significant time.

- **Update OLAP Services Cube caches to PRIME**: Select this check box to update all Intelligent Cubes to the PRIME architecture. This architecture can improve the performance of publishing Intelligent Cube results through the use of parallel processing and other optimizations. This update is applied to all Intelligent Cubes for the project.

- **Update data import cubes (definition and cache) to PRIME**: Select this check box to update all Data Import datasets to the PRIME architecture. This architecture can improve the performance of publishing Data Import results through the use of parallel processing and other optimizations. This update is applied to data sets included in your project through the use of Data Import.

  It is easier to update OLAP Services Cube caches and data import cubes to PRIME during the upgrade process. If you do not update them
now, each cube is updated when it is used. Doing the upgrade all at once ensures consistent performance for end users.

8. Click **Next**.

9. If you selected the **Migrate History List messages** option, on the History List Target Database page, specify the necessary information for the History List database.

10. Click **Next**.

11. On the Summary page, review your upgrade choices and click **Finish**.

### 4. History List Database Repository

If you are using a database-based History List, when upgrading to MicroStrategy 2020 you must create the History List database tables in the test environment.

To Create the Tables in the History List Repository

1. Open the MicroStrategy Configuration Wizard.

2. Select **Metadata, History List and Statistics Repository Tables** and click **Next**.

3. Select the **History List Tables** check box and click **Next**.

4. From the **DSN** drop-down list, select the DSN for your History List repository.

   If a DSN for your History List repository does not exist, you can select **New** to open the Connectivity Wizard and create a new DSN.

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

   The database user you provide becomes the owner of all History List tables and objects. The database user is required to have the Select, Create, Insert, and Drop permissions.
6. After providing a valid user name and password, you can click **SQL Preview** to open the SQL Preview dialog box. This dialog box provides the SQL statements that will be executed on your data source to create the History List tables. Click **Close** once you are done reviewing the SQL statements to return to the Configuration Wizard.

If you use the advanced options to change the SQL script, you can click SQL Preview after selecting the new script to see an updated listing of the SQL statements that will be executed.

7. Click **Advanced**.

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

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

If you use a table prefix for your History List tables, you must also define this table prefix when you create a database instance to connect to the History List tables.

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.

10. Click **Next**.
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.

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

5. Updating the Enterprise Manager Project

Upgrading from MicroStrategy 9 to 2020 includes major enhancements and updates to the Enterprise Manager project; whereas upgrading from MicroStrategy 10 to 2020 does not have major enhancements and upgrades to the Enterprise Manager project. In both scenarios, you should use the latest versions of Enterprise Manager projects with MicroStrategy 2020.

To retain customizations created in previous Enterprise Manager projects, make separate copies of these customized copies of the out-of-the-box objects, or as brand-new objects. Any modifications you made to the out-of-the-box objects for an Enterprise Manager project are replaced as part of the upgrade process.

- To upgrade your Enterprise Manager statistics and warehouse, ensure your Enterprise Manager Data Loader service is pointing to your Statistics and Enterprise Manager Repository. For more information, see **KB483298**.

- Use the Project Duplication Wizard to make a backup of your existing Enterprise Manager project. For detailed information about using Project Duplication, see the *Managing Your Projects* chapter of the *System Administration Guide*.

1. Open Configuration Wizard.

2. Select **Upgrade existing environment to MicroStrategy Secure Enterprise**, and click **Next**.

3. Select **Upgrade Enterprise Manager Project**, and click **Next**.
4. Provide the following information:
   - **User Name**: Type the MicroStrategy user name that can access and administer the Enterprise Manager project.
   - **Password**: Type the password for the MicroStrategy user that can access and administer the Enterprise Manager project.

5. If you have an Enterprise Manager project package file (.mmp) to use instead of the default file, click **Advanced >>**. For the **Package location** field, click ... (the Browse button) to navigate to and select the package file.

6. Click **Next**.

7. Provide the following information:
   - **DSN**: Select the data source name for your statistics repository or 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.

8. Click **Next**.

9. Review the summary information.

   You can click **Save** to save the configuration as a response (.ini) file to upgrade your Enterprise Manager projects on other systems or to run silent configurations at a later time. For information on running the Configuration Wizard with a response file, see [*Using a Response File with Configuration Wizard*](#).

10. Click **Finish**.
6. Upgrade the Platform Analytics Project

As of 2019, you can upgrade your Platform Analytics project in the metadata of your connected Intelligence Server. Upgrading the project is recommended with each platform and update release in order to brings in the latest dossiers, attributes, metrics and reporting optimizations to the Platform Analytics project.

1. Open Configuration Wizard.

2. Select Upgrade existing environment to MicroStrategy Secure Enterprise, and click Next.

3. Select Upgrade Platform Analytics Project, and click Next.

4. Provide the following information:

   - **User Name**: Enter the MicroStrategy user name that can access the Intelligence Server.
     
     If this is your first time connecting to the MicroStrategy Intelligence Server, use the user name Administrator without a password.
     
     - **Password**: Enter the password for the MicroStrategy user that can access the Intelligence Server.

5. Choose the Platform Analytics Repository.

   Select the MySQL/PostgreSQL DSN for the Platform Analytics Repository that was used to create Platform Analytics.

6. Enter your **User Name** and **Password** for the DSN.

7. Click Next.

8. Click **Apply**. The Configuration Wizard automatically applies the following configuration files:
7. Migrate Platform Analytics Data to MicroStrategy Repository

The Platform Analytics Data Migration tool is used to help existing customers migrate their data from MySQL to the newly supported PostgreSQL repository. This tool can help migrate both new and old versions of MySQL dump files to the latest version of Platform Analytics.

Backup Prerequisites:

- `/MicroStrategy/install/PlatformAnalytics/PAConsumerConfig.yaml` populated with:

  warehouseDbConnection:
  
  - whHost: 127.0.0.1
  
  - whUser: root
  
  - whPasswd: encrypted_password
  
  - whPort: 3306
  
  - whDb: platform_analytics_wh

- `mysql-connector-java.jar` is present in `PlatformAnalytics/lib` directory.

- Disk space sufficient to hold a backup of your MySQL `platform_analytics_wh` database.
**store Prerequisites:**

- **PAConsumerConfig.yaml populated with:**
  
  ```yaml
  pgWarehouseDbConnection:
  ```

  - **pgWhHost:** 127.0.0.1
  - **pgWhUser:** postgres
  - **pgWhPasswd:** encrypted password
  - **pgWhPort:** 5432
  - **pgWhDb:** platform_analytics_wh

- **Path to .csv files from a previous backup of platform_analytics_wh.**

- **Enough disk space available to PostgreSQL to restore the backed up ..csv files from MySQL.**

### Launching the Platform Analytics Data Migration Tool

1. **Navigate to your Platform Analytics home directory and go into the bin directory:**

   ```bash
   /opt/mstr/MicroStrategy/PlatformAnalytics/bin
   ```

2. **Run the following script:**

   ```bash
   ./platform-analytics-data-migration-tool.sh
   ```

3. **You will then be prompted with the following:**

   ```
   This is the Platform Analytics Data Migration Tool. The purpose of this tool is to help migrate your data from an existing Mysql Warehouse to a new PostgreSQL Warehouse.
   Please select from the following options:
   1) Backup
   2) Restore
   3) Backup and Restore
   0) Exit
   ```
Migration Workflow

Backup

1. Provide the path to the directory where the MySQL backup will be stored.

2. The tool will then begin backing up the MySQL platform_analytics_wh specified in your PAConsumerConfig.yaml file, placing the backup in your specified path.

Restore

1. Provide the path to the directory where the MySQL backup is stored.

2. The tool will prompt you again if you are sure you are okay to drop your PostgreSQL platform_analytics_wh schema.

3. If yes is selected, the platform_analytics_wh schema will be dropped and recreated matching the version of your MySQL dump.

4. The backup data is then imported into the newly created platform_analytics_wh schema.

5. The platform_analytics_wh schema will then be upgraded to the latest version of Platform Analytics.

Recommended Upgrade Procedures

1. On your new MicroStrategy 2020 machine, populate the PAConsumerConfig.yaml has the MySQL and PostgreSQL information shown in the prerequisites above.

2. Copy the mysql-connector-java.jar from your previous installation to the PlatformAnalytics/lib directory on the new machine.
3. Go to your PlatformAnalytics/bin directory and call the platform-analytics-data-migration-tool.sh file.

4. Select the Backup and Restore option (3).

5. Enter the full desired directory path for the database to be backed up to and restored from.

6. Wait until the backup is complete. The tool will then prompt if it is okay to recreate the PostgreSQL warehouse and select yes.

7. The program will then restore your MySQL backup files into your new PostgreSQL warehouse and the data migration will be complete.

8. Migrate Comments to a New Installation of Collaboration Server

In MicroStrategy 2020, Collaboration Server uses MicroStrategy Repository to store user comments. When performing a parallel upgrade, the new installation of Collaboration Server will be automatically configured to MicroStrategy Repository during installation. After performing the upgrade, Collaboration Server will only support connections to the MicroStrategy Data Repository. Any existing user comments will need to be migrated to the new repository to remain accessible.

The Collaboration Server Administration Tool provides a way for system administrators to migrate comments from the previous data repository to the new data repository. The Collaboration Administration Page will display a warning message when it detects that a previous data repository was used by the Collaboration Server before the upgrade.

The Collaboration Server Administration Tool is a console application and will connect to the data repository using the same configuration as the Collaboration Server so no user input is required to connect. The tool will provide feedback as data is being migrated and will show a message when the migration is complete.
On the existing Collaboration Server:

1. Edit the mongod.cfg file under the
   /opt/mstr/MicroStrategy/install/CollaborationServer/
   MongoDB/ directory, add a new section net with bindIpAll: true. It will allow the MongoDB to be accessible outside of the machine so we can use the Collaboration Server Administration Tool to connect to the MongoDB from another machine.

   Make sure to use four spaces instead of a tab for the indentation because the mongod.cfg is in YAML format.

   ```yaml
   ...
   processManagement:
     windowsService:
       serviceName: MSTR_mongodb
       displayName: MongoDB
   security:
     authorization: enabled
   net:
     bindIpAll: true
   ...
   ```

2. Restart the MongoDB service to apply the modified configuration.

   ```bash
   ./mongod.sh restart
   ```

3. Open the config.json file under the
   /opt/mstr/MicroStrategy/install/CollaborationServer/
   directory, copy the entire dburl property.

On the new Collaboration Server machine:


   ```bash
   ./collaborationServer.sh stop
   ```

2. Open the config.json file under the
/opt/mstr/MicroStrategy/install/CollaborationServer/directory paste the dburl line at the top of the content. Replace the localhost string in the dburl with the IP of the existing Collaboration Server machine. Make sure the content is in valid JSON format. Here is an example of the modified config.json file.

```
{
    "dburl": "mongodb://mstr:xxxx@[IP of the existing 2019 machine]:27017/mstr_collab?authSource=admin",
    "port": 3000,
    "logging": false,
    "authorizationServerUrl": "http://localhost:8080/MicroStrategyLibrary/api",
    "dataSource": {
        "username": "mstr_collab",
        "password": "xxxxxx"
    },
    "scaling": "none",
    "secretKey": "xxxx",
    "enableConfigApi": true
}
```

3. Run the Data Migration Command:

- Enter the three file locations in the following format:

  `$ "<PATH_TO_NODE>" "<PATH_TO_ADMIN_TOOL_FILE>" "<PATH_TO_CONFIG_FILE>"`

- Use the migrate command to run the data migration:
cmd> migrate

- Enter `quit` to close the Collaboration Server Administration Tool.


`./collaborationServer.sh start`

8. Activate Your Installation

All MicroStrategy 2020 installations must be activated within 30 days. Once you receive the activation code you will need to enter it into MicroStrategy License Manager.

1. Launch MicroStrategy License Manager:

   - **Command line**: In a Linux console window, browse to `/opt/mstr/MicroStrategy/bin` and run:
     
     `./mstrlicmgr -console`

   - **GUI**: In a Linux console window, browse to `/opt/mstr/MicroStrategy/bin` and run:
     
     `./mstrlicmgr`

2. Update your license key:

   - **Command line**:
     
     1. Choose option 2. **Update local License key**.
     
     2. Enter your MicroStrategy 2020 license key when prompted and press Enter.

   - **GUI**
     
     1. Open the **License Administration** tab > **Activate Installation** and click Next.
2. Select **Server Activation Using Code** and enter your activation code in the text box. Click **Next**

3. Click **OK**.

---

**Using MicroStrategy Backup and Restore**

MicroStrategy Backup is an application that allows you to easily back up essential MicroStrategy configurations, databases, and files. It also allows you to back up on-premises environments and restore your information to new MicroStrategy Cloud Platform for AWS or MicroStrategy Cloud Platform for Azure environments.

Upgrading with MicroStrategy Backup allows users to create mirror environments to test new projects, create disaster recovery environments, migrate current MicroStrategy deployment to the cloud, and more.

**How To Perform an Upgrade with MicroStrategy Backup**

1. **MicroStrategy Backup Preparations**

2. **Install MicroStrategy Backup**

3. **Run MicroStrategy Backup**

4. **Restore MicroStrategy Backup**

1. **MicroStrategy Backup Preparations**

Before you begin the upgrade with MicroStrategy Backup, take care of the following item:

- Intelligence Server must run on a Linux Operating System (preferably Red Hat Enterprise Linux or equivalent).

- Install Pigz Package on your environment.

Use the following command:

```
sudo yum install pigz -y
```
If you intend to restore an archive onto MicroStrategy Cloud Platform for AWS or MicroStrategy Cloud Platform for Azure:

- Projects must be unlocked if you intend to restore an archive onto MicroStrategy Cloud Platform for AWS or MicroStrategy Cloud Platform for Azure.

By default, projects are unlocked. To unlock a project:

1. Open Developer.

   If you are running MicroStrategy Developer on Windows for the first time, run it as an administrator.

   Right-click the program icon and select Run as Administrator. This is necessary in order to properly set the Windows registry keys. For more information, see KB43491.

2. From the folder list, select a project name.

3. In the menu bar, click Administration > Locking > Unlock.

Scheduler must be turned on.

Steps to enable Scheduler:

1. Open Developer.

   If you are running MicroStrategy Developer on Windows for the first time, run it as an administrator.

   Right-click the program icon and select Run as Administrator. This is necessary in order to properly set the Windows registry keys. For more information, see KB43491.

2. Right-click a project source and select Configure MicroStrategy Intelligence Server.
3. Under Server Definition, select **Advanced**.

4. Select the Use MicroStrategy Scheduler check box.

- Switching from an Enterprise environment to a Team or Department environment only allows you to choose a maximum of two nodes to back up cubes and caches. Be aware of the differences in RAM usage when moving from a different environment.

- Ensure you have an S3 bucket or Azure Storage Account (with Standard Performance Type) set up.

Be aware of the restrictions of MicroStrategy Backup:


- Additional flat files will be restored to a single directory. The back up and restoration of files requires your discretion.

- Additional DSNs require the same credential and server location as the metadata.

- You can only back up a maximum of four nodes in a cluster.

- You cannot back up OS configurations, such as timezone and heap size.

Once you have the above settings and installations, you can begin **installing MicroStrategy Backup**.

**2. Install MicroStrategy Backup**

To install MicroStrategy Backup, follow the instructions below:

1. Create a new directory for MicroStrategy Backup in your MicroStrategy Installation directory using the command:

   ```bash
   mkdir mstrbak
   ```
2. Download the MicroStrategy Backup Tool using the command:

```
wget https://s3.amazonaws.com/mstrbak/internal/mstrbak.zip
```

3. Unzip and extract the file.

4. Change the directory to the MicroStrategy Backup client executable using the following command:

```
cd Mstrbak-client
```

5. Run the following command in the MicroStrategy Backup directory to ensure the executable permissions are set for the files.

```
chmod +x fix-permissions.sh
```

Now that MicroStrategy Backup is downloaded and you are in the correct directory, you can move on to 3. Run MicroStrategy Backup.

### 3. Run MicroStrategy Backup

You are now positioned to run the MicroStrategy Backup client. You can run MicroStrategy Backup in one of three modes and, depending on the mode you select, running the MicroStrategy Backup client will differ.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Flag</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>How to Run in Promted Mode</strong></td>
<td>How - P</td>
<td>Runs the client in Prompted mode. This mode asks the user for information about the location of various MicroStrategy items. This is an interactive mode for users running MicroStrategy Backup for the first time.</td>
</tr>
<tr>
<td><strong>How</strong></td>
<td>-</td>
<td>Runs the client in Cloud Only mode. This flag indicates that the system</td>
</tr>
</tbody>
</table>
Mode Flag

you are running the client on was originally provisioned by the MicroStrategy Cloud Platform. This flag is a simplified version of prompted mode, and will only ask a subset of questions limited to credentials and storage location (S3 Bucket or Azure Storage Blob).

This is an express backup mode specifically for users with MSTR Cloud environments. This mode is ideal for users who want to quickly backup their entire environment in the default location. This mode is also ideal for upgrading by creating a parallel environment.

What are the default install locations?

<table>
<thead>
<tr>
<th>Value</th>
<th>Default File Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>BacFileLoc</td>
<td>/opt/mstr/mstrbak/</td>
</tr>
<tr>
<td>CachePath</td>
<td>/opt/mstr/MicroStrategy/IntelligenceServer/Caches/cloud_10s/</td>
</tr>
<tr>
<td>CubePath</td>
<td>/opt/mstr/MicroStrategy/IntelligenceServer/Cube/cloud_10s/</td>
</tr>
<tr>
<td>InboxPath</td>
<td>/opt/mstr/MicroStrategy/IntelligenceServer/Inbox/cloud_10s/</td>
</tr>
<tr>
<td>LibraryImagesPath</td>
<td>/opt/apache/tomcat/latest/webapps/MicroStrategyLibrary/images/</td>
</tr>
<tr>
<td>LibraryPluginsPath</td>
<td>/opt/apache/tomcat/latest/webapps/MicroStrategyLibrary/plugins/</td>
</tr>
<tr>
<td>MSTRInstalPath</td>
<td>/opt/mstr/MicroStrategy/</td>
</tr>
<tr>
<td>MobileConf</td>
<td>/opt/apache/tomcat/latest/webapps/MicroStrategy/</td>
</tr>
</tbody>
</table>
### How to Run in Silent Mode

Runs the client in Silent mode. This reads from a JSON file for information about the location of various MicroStrategy items. This mode is for automated processes or development operations. This mode works as a back up client in case you need to restore your information.

<table>
<thead>
<tr>
<th>Mode Flag</th>
<th>Value</th>
<th>Default File Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>igPath</td>
<td>rategyMobile/WEB-INF/xml/mobile/</td>
<td></td>
</tr>
<tr>
<td>MobileImagesPath</td>
<td>/opt/apache/tomcat/latest/webapps/MicroStrategyMobile/images/</td>
<td></td>
</tr>
<tr>
<td>MobilePluginsPath</td>
<td>/opt/apache/tomcat/latest/webapps/MicroStrategyMobile/plugins/</td>
<td></td>
</tr>
<tr>
<td>MySQLbin</td>
<td>/usr/bin/mysqldump</td>
<td></td>
</tr>
<tr>
<td>TomcatPath</td>
<td>/opt/apache/tomcat/latest/</td>
<td></td>
</tr>
<tr>
<td>WebConfigPath</td>
<td>/opt/apache/tomcat/latest/webapps/MicroStrategy/WEB-INF/xml/sys_defaults.properties</td>
<td></td>
</tr>
<tr>
<td>WebImagesPath</td>
<td>/opt/apache/tomcat/latest/webapps/MicroStrategy/images/</td>
<td></td>
</tr>
<tr>
<td>WebPluginsPath</td>
<td>/opt/apache/tomcat/latest/webapps/MicroStrategy/plugins/</td>
<td></td>
</tr>
</tbody>
</table>

### How to Run in Prompted Mode

Prompted mode is an interactive mode that asks you for information about your MicroStrategy environment, such as the path to the installation directory, cubes, caches, images, plugins, etc. Prompts beginning with
[REQUIRED] must be answered. You can skip prompts beginning with [OPTIONAL].

1. Enter the following command to run prompted mode. See Additional Flags for additional parameters to add to your execution command.

   ```bash
   ./mstrbak -p
   ```

2. Answer the prompted questions. To find the correct input for each entry, see How to Obtain MicroStrategy Backup Inputs.

   Prompted mode questions:

   1. Enter the install path `/opt/mstr/MicroStrategy`.

   2. Enter the project source name.

   3. Enter your credentials from your cloud environment's welcome email.

   4. Enter the name of the metadata DSN.

   5. Enter your credentials from your cloud environment's welcome email.

   6. Enter the path to the dump bin.

   7. Enter your metadata prefix.

   8. Enter `yes` to keep the same projects loaded.

      Enter `no` if you don't want to keep your projects loaded and enter the names of the projects to register and unregister.

   9. Enter `yes` to backup your Platform Analytics warehouse and enter your Platform Analytics DSN name.

      Enter `no` if you don't want to backup your Platform Analytics warehouse.

   10. Enter `yes` to backup your Collaboration Server database and
enter your server definition.

Enter no if you don't want to backup your Collaboration Server database.

11. Enter yes to backup your History List. You will be asked where your history list is stored.
   - Enter 1 for Metadata
   - Enter 2 for External warehouse
   - Enter 3 for File.
   Enter no if you don't want to backup your History List.

12. Enter yes to backup additional warehouses.
    Enter no if you don't want back up any additional warehouses.

13. Enter yes to backup any of your cubes and enter the full path to your cubes.
    Enter no if you don't want to backup your cubes.

14. Enter yes to backup any of your caches and enter the full path to your caches.
    Enter no if you don't want to backup your caches.

15. Enter yes to backup the WSRM and enter the full path to the WSRM.
    Enter no if you don't want to backup the WSRM.

16. Enter yes to backup your MicroStrategy Web, Library, and Mobile images. The following questions appear:
- Enter the path to MicroStrategy Web.
- Enter the path to MicroStrategy Library.
- Enter the path to MicroStrategy Mobile.

Enter no if you don't want to backup your images.

17. Enter yes to backup your MicroStrategy Web, Library, and Mobile plugins. The following questions appear:
- Enter the path to MicroStrategy Web.
- Enter the path to MicroStrategy Library.
- Enter the path to MicroStrategy Mobile.

Enter no if you don't want to backup your plugins.

18. Enter yes to backup your MicroStrategy Web, Library, and Mobile configurations.

Enter no if you don't want to backup your configurations.

19. Enter yes to backup additional flat files and enter the file path(s) separated by commas. Additional files are backed up, but not restored. During restore, additional files are stored in the opt/mstr/mstrbak/additional_files directory.

Enter no if you don't want to backup additional flat files.

20. Enter yes to upload your backed up files to cloud storage. You will be asked which cloud storage option to upload to.
- Enter 1 to upload to your AWS S3 Bucket and enter the S3 bucket name. If prompted, provide the S3 access keys and secret key.
3. The client runs and prints messages to the Terminal as it proceeds.

4. The execution is complete when the following message appears:

```
MSTRBak has finished successfully.
Archive created locally at:
path/to/backup/ARCHIVE.tar.gz
Archive uploaded to S3 at:
https://link/to/ARCHIVE.tar.gz
```

Your MicroStrategy Backup is located in the directory listed in the message and in specified S3 bucket or Azure Blob, if selected. A silent response JSON file is created at the conclusion of the execution, even if the backup was not successful. You may use this response file to run all subsequent executions in Silent mode.

Additional Flags

Append any of the following flags to the execution command as desired.

- To show verbose logging, enter the parameter `-v`.

- To dump the database, enter the parameter `-d`. It is not recommended to use this parameter if you plan to use MicroStrategy Backup to upgrade. The `-d` parameter should be used when backing up MicroStrategy 2019 and restoring on MicroStrategy 2019.

- To receive email notifications about the backup process, enter the parameter `-m <your email>`.

For example,
How to Obtain MicroStrategy Backup Inputs

The following list details where you can locate answers to prompted questions.

<table>
<thead>
<tr>
<th>Input</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>BakFileLocation</td>
<td>This is any location on the Linux machine where the file is written.</td>
</tr>
<tr>
<td>MSTRInstallPath</td>
<td>By default, on cloud environments it is /opt/mstr/MicroStrategy/.</td>
</tr>
<tr>
<td></td>
<td>If this is not the correct path, you can find the path by running locate MSIReg.reg as this file will only exist in the MicroStrategy install path.</td>
</tr>
<tr>
<td>ServerDef</td>
<td>Open the MSIReg.reg file in your text editor and search for ServerInstanceName.</td>
</tr>
<tr>
<td>CubePath</td>
<td>Open Project Configuration for one of your projects and choose Intelligent Cubes &gt; General. The path appears in the top window beginning with MSTRInstallPath. For example, ./Cube/env-31097laiouse1 = /opt/mstr/MicroStrategy/Cube/env-laiouse1</td>
</tr>
<tr>
<td>CachePath</td>
<td>Open Project Configuration for one of your projects and choose Caching &gt; Result Caches &gt; Storage. The path appears in the top window beginning with MSTRInstallPath. For example, ./Caches/env-31097laiouse1 = /opt/mstr/MicroStrategy/Caches/env-laiouse1</td>
</tr>
</tbody>
</table>
### Upgrade Guide

<table>
<thead>
<tr>
<th>Input</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobileConfigPath</td>
<td>Most often it will be in /tomcat/{version}/webapps/MicroStrategyMobile/WEB-INF/xml/mobile.</td>
</tr>
<tr>
<td>WebConfigPath</td>
<td>Most often it will be in /tomcat/{version}/webapps/MicroStrategy/WEB-INF/xml/sys_defaults.properties.</td>
</tr>
<tr>
<td>LibraryImagesPath</td>
<td>Most often it will be in /tomcat/{version}/webapps/MicroStrategyLibrary/images.</td>
</tr>
<tr>
<td>MobileImagesPath</td>
<td>Most often it will be in /tomcat/{version}/webapps/MicroStrategyMobile/images.</td>
</tr>
<tr>
<td>WebImagesPath</td>
<td>Most often it will be in /tomcat/{version}/webapps/MicroStrategy/images.</td>
</tr>
<tr>
<td>LibraryPluginsPath</td>
<td>Most often it will be in /tomcat/{version}/webapps/MicroStrategyLibrary/plugins.</td>
</tr>
<tr>
<td>MobilePluginsPath</td>
<td>Most often it will be in /tomcat/{version}/webapps/MicroStrategyMobile/plugins.</td>
</tr>
<tr>
<td>WebPluginsPath</td>
<td>Most often it will be in /tomcat/{version}/webapps/MicroStrategy/plugins.</td>
</tr>
<tr>
<td>MySQLbin</td>
<td>Run the command which mysql dump as root.</td>
</tr>
<tr>
<td>MongoDBbin</td>
<td>Run the command which mongodump as root.</td>
</tr>
<tr>
<td>MetadataDSN</td>
<td>Open the MSIReg.reg file and search Location. Use the location in [HKEY_LOCAL_MACHINE\SOFTWARE\MicroStrategy\Data Sources\CastorServer].</td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td><strong>Location</strong></td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HistoryListDSN</td>
<td>Open the Intelligence Server Configuration in Developer. Take note of the Database Instance name. Close the Intelligence Server Configuration and choose Administration &gt; Configuration Managers &gt; Database Instances. Right-click and edit the database instance you found in the Intelligence Server Configuration. Click <strong>Modify</strong>. The HistoryListDSN is the name of the ODBC data source in the Database Connections window.</td>
</tr>
<tr>
<td>StatisticsDSN</td>
<td>Open Project Configuration for one of your projects that is logging statistics. Choose Database Instances &gt; Statistics. Take note of the database instance name.</td>
</tr>
<tr>
<td>MySQLUser</td>
<td>This is the username of an administrator user in the database.</td>
</tr>
<tr>
<td>MySQLPwd</td>
<td>This is the password of an administrator user in the database.</td>
</tr>
<tr>
<td>MSTRUser</td>
<td>The username of a metadata user with full administrative privileges (Administrator is recommended).</td>
</tr>
<tr>
<td>MSTRPwd</td>
<td>The password of a metadata user with full administrative privileges (Administrator is recommended).</td>
</tr>
<tr>
<td>AdditionalDSNs</td>
<td>List the names of any additional DSNs that will be needed in the new environment. These names can be found in the odbc.ini file located in the MSTRInstallPath.</td>
</tr>
<tr>
<td>ProjectsToRegister</td>
<td>For all projects, enter *. For multiple projects, add brackets with each project name in quotations separated by commas.</td>
</tr>
<tr>
<td></td>
<td><strong>For example</strong>, [&quot;MicroStrategy Tutorial&quot;, &quot;Enterprise Manager&quot;]</td>
</tr>
<tr>
<td>ProjectsToUnregister</td>
<td>For all projects, enter *. For multiple projects, add</td>
</tr>
<tr>
<td>Input</td>
<td>Location</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>brackets with each project name in quotations separated by commas.</td>
<td></td>
</tr>
<tr>
<td><strong>For example,</strong> [“MicroStrategy Tutorial”, “Enterprise Manager”]</td>
<td>A Main Project is the project that will be default to when clicking <strong>Create Dossier</strong> on the new cloud landing page.</td>
</tr>
<tr>
<td><strong>This can be any of the projects in the ProjectsToRegister section.</strong></td>
<td><strong>Open License Manager</strong> on the Intelligence Server machine and open <strong>Audit</strong>. Note the project source that connects the Intelligence Server. If there is not a project source, create one using the Configuration Wizard.</td>
</tr>
<tr>
<td><strong>Enter true or false. Only enter true if the old environment is a clustered environment.</strong></td>
<td><strong>If ClusteredCubes is true, then this section will be a list of the Intelligence Servers in brackets with each server name in quotations separated by commas.</strong></td>
</tr>
<tr>
<td><strong>For example,</strong> [“env-1000”, “env-2000”]</td>
<td><strong>Enter Yes or No. Enter Yes to upload the MicroStrategy Backup file to AWS S3 or Azure Blob.</strong></td>
</tr>
<tr>
<td><strong>This is the name of the S3 bucket (without S3://).</strong></td>
<td><strong>The S3 access key used to access the S3 bucket.</strong></td>
</tr>
<tr>
<td><strong>The S3 secret key used to access the S3 bucket.</strong></td>
<td><strong>This is the name of the Storage Account in Azure.</strong></td>
</tr>
<tr>
<td><strong>The Access Key used to access the Storage Account in Azure.</strong></td>
<td></td>
</tr>
</tbody>
</table>
How to Run in Cloud Mode

Cloud mode is an express backup mode specifically for users with MicroStrategy cloud environments. This mode is ideal for users who want to quickly back up their entire environment. Cloud mode can handle all enterprise applications.

1. Enter the following command to run cloud mode. See Additional Flags for additional parameters to add to your execution command.

   ```bash
   ./mstrbak -c
   ```

2. Enter your credentials from your cloud environment's welcome email.

   Your backup file is located in `/opt/mstr/mstrbak/` and your response file is located in `/opt/mstr/mstrbak/mstrbakResponseFile.json`.

3. Enter your credentials to the database.

4. Enter the SQL dump bin path.

5. Enter `y` to upload to a cloud storage. When prompted, choose which cloud storage option to upload to. Enter `n` if you don't want to upload to a cloud storage and complete the backup.

   - Enter `1` to upload to your AWS S3 Bucket and enter the S3 bucket name. If prompted, provide the S3 access keys and secret key.
   - Enter `2` to upload to your Azure Blob Storage and enter your Storage Account Name and Container Name. If prompted, provide the Storage Account access key.

6. The client runs and prints messages to the Terminal as it proceeds.

7. The execution is complete when the following message appears:

   ```text
   MSTRBak complete. Operation successful? True
   Archive created locally at:
   path/to/backup/ARCHIVE.tar.gz
   ```
Archive uploaded to S3 at:
https://link/to/ARCHIVE.tar.gz

The client in cloud mode will attempt to backup all DSNs in the odbc.ini. If some data warehouses fail to backup, an error message appears and operation success? reads as false. However, if the DSNs are successfully backed up, you can use the backup package.


Additional Flags

Append any of the following flags to the execution command as desired.

- To show verbose logging, enter the parameter -v.
- To dump the database, enter the parameter -d. It is not recommended to use this parameter if you plan to use MicroStrategy Backup to upgrade.
- To receive email notifications about the backup process, enter the parameter -m <your email>.

For example,

./mstrbak -c -v -d -m johnSmith@example.com -m janeDoe@example.com
How to Run in Silent Mode

Silent mode allows you to specify a JSON file with prepared responses for information pertaining to the MicroStrategy environment. Before running, you need to create a silent response file.

1. Create a response file based on the following template. See Response File Schema Template for schema reference.

```json
{
    "mstrbak": {
        "mstrbak_version": 1.0,
        "backup_path": "/opt/mstr/mstrbak",
        "response_file_path": "/opt/mstr/mstrbak/response.json"
    },
    "mstr": {
        "mstr_version": "11.1.000.123",
        "install_path": "/opt/mstr/MicroStrategy",
        "driver_paths": {
            "mysql": "/usr/share/java/mysql-connector-java-8.0.12.jar"
        },
        "hostnames": ["env-12345laiouse1"],
        "username": "mstr",
        "password": "",
        "history_list_storage": "external",
        "project_source_name": "mstr_metadata",
        "server_definition": "cloud_10s",
        "cube_path": "/opt/mstr/MicroStrategy/IntelligenceServer/Cube/cloud_10s",
        "cache_path": "/opt/mstr/MicroStrategy/IntelligenceServer/Caches/cloud_10s",
        "inbox_path": "/opt/mstr/MicroStrategy/IntelligenceServer/Inbox/cloud_10s",
        "wsrm_path": "",
        "projects": {
            "main_project": "MicroStrategy Tutorial",
            "projects_to_register": [
                "MicroStrategy Tutorial"
            ]
        }
    }
}
```
"projects_to_unregister": [
]
"projects_locked": [
]
"configuration_locked": false,
"scheduler_on": true
"collaboration_server": {
"backup": false,
"db_type": 
"username": 
"password": 
"bin_path": 
"backup": false,
"dsn_name": 
"db_type": 
"username": 
"password": 
"bin_path": 
"backup": true,
2. For the `history_list_storage` parameter, enter metadata, external, or file.

3. For the `db_type` parameter, enter MySQL, PostgreSQL, or MongoDB.

4. Save the JSON file, ensure the file extension is .json.

5. (Optional) Create a backup of the newly created JSON file.

   This is recommended because, after the Silent Mode execution, the passwords in the response file are hidden for security reasons. To avoid losing any passwords, keep a back up of the JSON file.

6. Enter the following command to run silent mode. See Additional Flags for additional parameters to add to your execution command.

   ```bash
   ./mstrbak -r /path/to/response.json
   ```
7. The client runs and prints messages to the Terminal as it proceeds.

8. The execution is complete when the following message appears:

```
MSTRBak complete. Operation successful? True
Archive created locally at:
path/to/backup/ARCHIVE.tar.gz
Archive uploaded to S3 at:
https://link/to/ARCHIVE.tar.gz
```

Your MicroStrategy Backup archive is located in the directory listed in the message and in the specified S3 bucket or Azure Blob, if selected.


**Additional Flags**

Append any of the following flags to the execution command as desired.

- To show verbose logging, enter the parameter `–v`.
- To dump the database, enter the parameter `–d`. It is not recommended to use this parameter if you plan to use MicroStrategy Backup to upgrade.
- To receive email notifications about the backup process, enter the parameter `–m <your email>`.

For example,

```
./mstrbak -r /path/to/response.json -v -d -m johnSmith@example.com -m janeDoe@example.com
```
Response File Schema Template

The following is a template for response file schema. This template is to be used in tandem with the JSON file template on *How to Run in Silent Mode*

```json
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "mstrbak": {
      "type": "object",
      "properties": {
        "mstrbak_version": {
          "type": "number"
        },
        "backup_path": {
          "type": "string"
        },
        "response_file_path": {
          "type": "string"
        }
      }
    },
    "required": [
      "mstrbak_version",
      "backup_path",
      "response_file_path"
    ]
  }
},

"mstr": {
  "type": "object",
  "properties": {
    "mstr_version": {
      "type": "string"
    },
    "install_path": {
      "type": "string"
    },
    "driver_paths": {
      "type": "object",
      "properties": {
    ...
```
"mysql": {
  "type": "string"
},
"required": [
  "mysql"
],
"hostnames": {
  "type": "array",
  "items": [
    {
      "type": "string"
    }
  ]
},
"username": {
  "type": "string"
},
"password": {
  "type": "string"
},
"project_source_name": {
  "type": "string"
},
"server_definition": {
  "type": "string"
},
"cube_path": {
  "type": "string"
},
"cache_path": {
  "type": "string"
},
"history_list_storage": {
  "type": "string"
},
"inbox_path": {
  "type": "string"


```
}
"projects": {
  "type": "object",
  "properties": {
    "main_project": {
      "type": "string"
    },
    "projects_to_register": {
      "type": "array",
      "items": [
        {
          "type": "string"
        }
      ]
    },
    "projects_to_unregister": {
      "type": "array",
      "items": {}
    },
    "projects_locked": {
      "type": "array",
      "items": {}
    }
  },
  "required": [
    "main_project",
    "projects_to_register",
    "projects_to_unregister",
    "projects_locked"
  ]
},
"configuration_locked": {
  "type": "boolean"
},
"scheduler_on": {
  "type": "boolean"
}
"required": [
```
"mstr_version",
"install_path",
"driver_paths",
"hostnames",
"username",
"password",
"history_list_storage",
"project_source_name",
"server_definition",
"cube_path",
"cache_path",
"inbox_path",
"hybrid_path",
"wsrm_path",
"projects",
"configuration_locked",
"scheduler_on"
]}

"collaboration_server": {
  "type": "object",
  "properties": {
    "backup": {
      "type": "boolean"
    },
    "db_type": {
      "type": "string"
    },
    "username": {
      "type": "string"
    },
    "password": {
      "type": "string"
    },
    "bin_path": {
      "type": "string"
    }
  },
  "required": [
"backup",
"db_type",
"username",
"password",
"bin_path"
]
},
"dsns": {
"type": "object",
"properties": {
"metadata": {
"type": "object",
"properties": {
"prefix": {
"type": "string"
},
"dsn_name": {
"type": "string"
},
"db_type": {
"type": "string"
},
"username": {
"type": "string"
},
"password": {
"type": "string"
},
"bin_path": {
"type": "string"
}
},
"required": ["prefix",
"dsn_name",
"db_type",
"username",
"password",
"bin_path"
"platform_analytics": {
  "type": "object",
  "properties": {
    "backup": {
      "type": "boolean"
    },
    "dsn_name": {
      "type": "string"
    },
    "db_type": {
      "type": "string"
    },
    "username": {
      "type": "string"
    },
    "password": {
      "type": "string"
    },
    "bin_path": {
      "type": "string"
    }
  },
  "required": [
    "backup",
    "dsn_name",
    "db_type",
    "username",
    "password",
    "bin_path"
  ]
},
"history_list": {
  "type": "object",
  "properties": {
    "backup": {
      "type": "boolean"
    }
  }
}
"dsn_name": {
    "type": "string"
},
"db_type": {
    "type": "string"
},
"username": {
    "type": "string"
},
"password": {
    "type": "string"
},
"bin_path": {
    "type": "string"
},
"required": [
    "backup",
    "dsn_name",
    "db_type",
    "username",
    "password",
    "bin_path"
],
"additional_dsn": {
    "type": "array",
    "items": [
        {
            "type": "object",
            "properties": {
                "backup": {
                    "type": "boolean"
                },
                "dsn_name": {
                    "type": "string"
                },
                "username": {
                    "type": "string"
                }
            }
        }
    ]
}
},
  "password": {
    "type": "string"
  },
  "bin_path": {
    "type": "string"
  }
},
  "required": [
    "backup",
    "dsn_name",
    "username",
    "password",
    "bin_path"
  ]
}

},
  "required": [
    "metadata",
    "platform-analytics",
    "history_list",
    "additional_dsns"
  ]
},
  "library": {
    "type": "object",
    "properties": {
      "path": {
        "type": "string"
      },
      "backup_contents": {
        "type": "object",
        "properties": {
          "backup_images": {
            "type": "boolean"
          },
          "backup_plugins": {
            "type": "boolean"
          }
        }
      }
    }
  }
"type": "boolean"
},
"backup_configuration_files": {
  "type": "object",
  "properties": {
    "mapConfig.xml": {
      "type": "boolean"
    },
    "configOverride.properties": {
      "type": "boolean"
    }
  },
  "required": [
    "mapConfig.xml",
    "configOverride.properties"
  ]
},
"required": [
  "backup_images",
  "backup_plugins",
  "backup_configuration_files"
]
}
},
"required": [
  "path",
  "backup_contents"
]
},
"web": {
  "type": "object",
  "properties": {
    "path": {
      "type": "string"
    },
    "backup_contents": {
      "type": "object",
      "properties": {
        "backup_images": {
          "type": "boolean"
        },
        "backup_plugins": {
          "type": "boolean"
        },
        "backup_configuration_files": {
          "type": "boolean"
        }
      }
    }
  }
}
"backup_images": {
    "type": "boolean"
},
"backup_plugins": {
    "type": "boolean"
},
"backup_configuration_files": {
    "type": "object",
    "properties": {
        "sys_defaults.properties": {
            "type": "boolean"
        },
        "mapConfig.xml": {
            "type": "boolean"
        }
    },
    "required": [
        "sys_defaults.properties",
        "mapConfig.xml"
    ]
},
"required": [
    "backup_images",
    "backup_plugins",
    "backup_configuration_files"
]
},
"required": ["path", "backup_contents"]
},
"mobile": {
    "type": "object",
    "properties": {
        "path": {
            "type": "string"
        }
    }
}
"backup_contents": {
  "type": "object",
  "properties": {
    "backup_images": {
      "type": "boolean"
    },
    "backup_plugins": {
      "type": "boolean"
    },
    "backup_configuration_files": {
      "type": "object",
      "properties": {
        "WEB-INF/xml/mobile": {
          "type": "boolean"
        },
        "mapConfig.xml": {
          "type": "boolean"
        }
      },
      "required": [
        "WEB-INF/xml/mobile",
        "mapConfig.xml"
      ]
    }
  },
  "required": [
    "backup_images",
    "backup_plugins",
    "backup_configuration_files"
  ]
},
"aws": {
Upgrade Guide

"type": "object",
"properties": {
  "upload": {
    "type": "boolean"
  },
  "s3_bucket_name": {
    "type": "string"
  },
  "s3_access_key": {
    "type": "string"
  },
  "s3_secret_key": {
    "type": "string"
  }
},
"required": [
  "upload",
  "s3_bucket_name",
  "s3_access_key",
  "s3_secret_key"
],
"azure": {
    "type": "object",
    "properties": {
      "upload": {
        "type": "boolean"
      },
      "account_name": {
        "type": "string"
      },
      "account_key": {
        "type": "string"
      },
      "container_name": {
        "type": "string"
      }
    },
    "required": [}
4. Restore MicroStrategy Backup

MicroStrategy Backup is a MicroStrategy application that allows users to easily back up essential MicroStrategy configurations, databases, and files. The backed-up package created by the client can either be stored locally or they can be via the MicroStrategy Cloud Platform for AWS or MicroStrategy Cloud Platform for Azure with the MicroStrategy Backup restoration capability. Restoring your MicroStrategy Backup allows you to restore cubes, caches, ODBC DSNs, update projects, and more.

✔️ To restore MicroStrategy Backup in AWS, you must have uploaded your backup to an S3 bucket (directly inside the bucket and not inside any folder) in 3. Run MicroStrategy Backup.
To restore MicroStrategy Backup in Azure, you must have uploaded your backup to an Azure Storage account which can be accessed by the new environment in 3. Run MicroStrategy Backup.

1. Log in to the MicroStrategy Cloud Platform.

2. Provision a new environment as either Team, Department, or Enterprise.

3. Under Restore an Environment, select the Restore environment from MicroStrategy Backup check box.

4. Enter the S3 bucket or Azure Blob URL.

   The Backup file can only be restored to environments with the same or higher version numbers.

5. Click Validate.

6. Click Create Environment. Your MicroStrategy Backup is restored on your new environment.
Validating the Upgrade

It is essential that you fully test your new MicroStrategy system to see how it withstands the kind of typical day-to-day system usage your user community requires. If you have upgraded first in a test environment, you can run these tests in the test environment and correct any problems you may find there. Then when you upgrade your production environment, you already have an idea of the kinds of difficulties you may encounter. For information about setting up an upgrade test environment, see Configuring an Upgrade Test Environment, page 68.

During the test, make sure you are satisfied with the status and efficiency of your MicroStrategy system. If you need help, see the Readme or contact MicroStrategy Technical Support.

URL Scanner - Identify URLs for Export

As of MicroStrategy 2020 Update 1, administrators can specify which URLs or URL paths are permitted when fetching content to be included in an export. This concept, where only certain URLs are permitted, is largely referred to as whitelisting. To easily identify which URLs belong in this list after upgrading, it's recommended to use the URL Scanner.

The URL Scanner tool allows you to scan MicroStrategy metadatas (10.4.x and higher) for URLs accessed by the exporting workflows on the Intelligence Server. It lists all the URLs referenced by reports, documents, and dossiers and allows you to make an informed decision about which URLs are safe to be used in your environment. Combined with the new whitelisting functionality in MicroStrategy 2020 Update 1, this tool can be used to whitelist only those safe URLs to protect against Server Side Request Forgery attacks.

The recommended workflow for using the URL Scanner is as follows:
1. Review the prerequisites and system requirements.
2. Run the URL Scanner.
3. Review the scan results.
4. Add URLs and URL paths from the scan results to the whitelist.

Get Started

First, download the zip file from the MicroStrategy Download site and unzip it to a folder. We will refer to this as “URL Scanner Folder” going forward.

URL Scanner System Requirements

The URL Scanner supports all Microsoft Windows Systems (except Microsoft Vista) supported by MicroStrategy Developer.

The following systems are certified for use with this tool:

- Windows Server 2016 Enterprise
- Windows Server 2019 Enterprise
- Windows 10

MicroStrategy Prerequisites

- You must have the Bypass all object security access checks privilege to use the tool.
- You must connect through a 2-tier (direct) or 3-tier project source with Standard authentication mode.
- Your machine must have MicroStrategy Developer installed.

Compatibility

The URL Scanner is compatible with metadata versions 10.4.x and above.
Run the URL Scanner

You must perform the prerequisites on Get Started before proceeding.

1. Double-click the **URLScanner.exe** file.
2. Use the drop-down to choose an available project source. It's recommended to use a 2-tier project source.
3. Enter the username and password for the selected project source.
4. Click **Connect**.
5. In the project box, select the project(s) that you want to scan. It's recommended to choose the **Select all projects** checkbox.
6. Click **Scan**. After the URL scan has completed, a dialog appears with the directory of the output files. The output files are located in the URL Scanner Folder.
7. Click **Exit**. You can exit the scan at any point. Since the scan results are incrementally generated, you can still check the scan result if the scan is canceled.

Review Scan Results

The scan results are generated in the same folder as the URL Scanner executable.

The file name follows the pattern **URLs_YYMDD_TIMESTAMP.csv** and contains the following information:

- Object name
- Object path
- Object owner
- URL type
For example, a scan result could appear as:

"2.0 Book of Business","MERIT System\Public Objects\Reports\W. Work In Progress\Old Reports\2.0 Book of Business","Administrator","FieldImage","Content > Layout: Opportunity Overview","http://www.microstrategy.com/cmstemplates/microstrategy/images/microstrategy_logo.png"

Once you have the generated .csv file, add the URLs and URL paths to the Allow URLs to Export list.

Add URLs to Whitelist

Now that you have a list of URLs, you can add them to the Allow URLs to Export list.

Adding URLs or URL paths to this list permits the content to be fetched when exporting a report, document, or dossier to PDF or Excel.

To modify this field, you must have the Web Administration privilege, Configure Server Basic privilege, and read/write access for the server definition.

If you are using MicroStrategy Web ASP, please see Additional Step for MicroStrategy Web ASP Customers in KB484127 before proceeding.

Relative paths are case sensitive.

1. Log in to a project using an account with administrative privileges.
2. From the upper right of any page, click the username drop-down and select Preferences. The Preferences page opens.
3. From the pane on the left, select Project Defaults.
4. From the pane on the left, select Security. The Security project
defaults page opens.

5. From **Allow URLs to Export**, enter a URL or URL path.

   Use a relative path or the following protocols: https://, http://, or file:///.

6. Click **Add**.

7. Click **Apply**. This setting applies to all projects in your environment and affects all MicroStrategy products that have export functionality.

8. Clear your Export cache.

   1. In Developer, log in to a project source. You must have the Monitor Caches privilege.

   2. Expand **Administration > SystemMonitors > Caches > Documents**.

   3. Select a project and click **OK**.

   4. Locate caches with the format **PDF** or **Excel** and delete them.

Once you have whitelisted the URLs found in your metadata, it is recommended to perform **Integrity Manager testing** to validate the upgrade impact.

Thereafter, administrators should review any **URLs or URL paths** that are still blocked from export.

Related Articles

**KB484127: Securing PDF and Excel Export with Whitelists**

**Integrity Manager - Upgrade Impact Testing**

After upgrading your production environment, you should ensure that the changes involved do not alter any of your report, document, or dossier
results. If changes have occurred, you can understand why using MicroStrategy Integrity Manager.

MicroStrategy Integrity Manager is designed to execute and compare reports, documents, and dossiers from different versions by examining their data and SQL. Starting in 11.0, administrators can compare data down to the visualization level of all content objects. Changes found during comparison testing are logged automatically and can be viewed using the Upgrade Analysis Dossier. See *Upgrade Analysis Dossier* for more information.

For detailed information about Integrity Manager, including instructions, see the *Integrity Manager* chapter in the *System Administration Guide*.

**Best Practices for Integrity Testing**

MicroStrategy recommends the following best practices when testing your upgrade with Integrity Manager:

- Create all baselines using Integrity Manager 2020.

  For example, if you are going to upgrade from 10.4.x to 2020 and you will not have a live 10.4.x and 2020 both available at the same time to do a Project vs Project comparison, then upgrade a developer machine to 2020 with Integrity Manager to capture the baseline for 10.4.x before upgrading it to 2020. Baselines that are created with older versions of Integrity Manager are not compatible and administrators can expect “does not contain tags” or “does not contain relations” errors. This is due to intentional architecture changes to support new functionality.

- Create an integrity test comparing reports, documents, and dossiers from the upgraded test environment with the same objects in the current production environment, so that you can easily see where any differences are.

- In a comparative integrity test, you must have the same OS version and the same font installed on your machine to use the Graph view to compare
two PDF reports. Font rendering on a PDF is version and OS specific, so differences may result in formatting issues, which can affect comparison results.

- Execute the tests against the production data warehouse. If this is not possible, test against a data warehouse that resembles the production data warehouse as closely as possible. Ensure that the tests for both the upgrade environment and the current production environment are executed against the same warehouse or unmatched results are returned, which is expected.

- Execute the tests with production users, groups, and security roles, instead of specially-created integrity test users and groups, so that the tests match the production environment as closely as possible.

- If possible, select content objects with at least default prompt answers to test. Otherwise create static copies of content objects that have prompts with no default prompt answers.

- When executing a performance test, ensure no other usage or testing is being performed on the environments as it can result in misleading results.

Creating a Project Baseline

When you upgrade to MicroStrategy 2020, you must ensure that reports, report services documents, and dossiers return the same results in both the current environment and the upgraded environment. For customers performing a parallel upgrade, the easiest way to do this is to use the “Project versus Project” test which will execute selected content objects against two environments, one your existing production, and the other your test upgrade environment. Customers performing an in-place upgrade will need to create a project baseline of your current production environment, and then after the upgrade is done, use that baseline to run the “Baseline versus Project” option. The steps below provide an example.
Create a project baseline

1. In your current MicroStrategy environment, access the Windows desktop with MicroStrategy clients installed.

2. Open Start Menu > MicroStrategy Products > Integrity Manager.

3. From the File menu, select Create Test.

4. Select Single Project and click Next.

5. On the Enter Base Project Information page, enter the following details about your current environment:
   
   - In the Server Name field, enter the name or IP address of your Intelligence Server.
   
   - In the Server Port field, enter the port number your Intelligence Server uses. The default port number is 34952.
   
   - From the Authentication Mode field, select the authentication mode your Intelligence Server uses.
In the Login ID and Password fields, enter the credentials to log in MicroStrategy Intelligence Server.

From the Project drop-down list, select the project to test.

6. Click Next.

7. On the **Select Objects from the Base Project to be included in Test** page, select objects you want to test.

8. Click Next.

9. On the **Select Prompt Settings** screen, click Next.

10. On the **Select Execution Settings** screen, click Next.

11. On the **Select Processing Options** screen, check the following options for each type of objects:

   **Reports:**
   - SQL/MDX
   - Data
   - Graph
   - PDF

   **Documents:**
   - Execution
   - Data
   - PDF

   **Dossiers:**
   - Execution
   - Data
12. Click **Next**.

13. In the Summary screen, click **Run**.
14. In the Save window, click Yes.

15. Save the test and make note of the location. You will need the results to run the test on your new environment.

16. When the test executions are complete, exit Integrity Manager.

**Compare Project Baseline versus Upgraded Production**

After the upgrade is completed, your project baseline results to run a comparison with your new MicroStrategy 2020 environment.

1. Open **Start Menu > MicroStrategy Products > Integrity Manager**.

2. From the **File** menu, select **Create Test**.

3. Select **Baseline versus Project** and click **Next**.
4. On the Select Base screen, in the Select Baseline XML file field, select the baseline file you created in the previous Integrity Manager test.

5. Click Next.

6. On the Enter Target project Information page, enter the following details about your MicroStrategy 2020 environment:
   
   - In the Server Name field, enter the name of your 2020 Intelligence Server.
   
   - In the Server Port field, enter the port number your Intelligence Server uses. The default port number is 34952.
   
   - From the Authentication Mode field, select the authentication mode your Intelligence Server uses.
   
   - In the Login ID and Password fields, enter the credentials to log in MicroStrategy Intelligence Server.
1. From the Project drop-down list, select the project you used in the first test.

7. On the **Select Contents to Test** page, select the objects you used in the first test.

8. Click **Next**.

9. On the **Select Prompt Settings** screen, keep all of selected options, click **Next**.

10. On the **Select Execution Settings** screen, keep all of selected options, click **Next**.

11. On the **Select Processing Options** screen, keep all of selected options, click **Next**.

12. In the Summary screen, click **Run**.

13. In the **Save** window, click **Yes** and save the test.

14. When the test executions are complete, exit Integrity Manager and proceed to understanding your results using the Upgrade Analysis Dossier.

---

**MicroStrategy Capacity Testing Tool**

**Capabilities and Restrictions**

The MicroStrategy Capacity Testing Tool is a client application which connects to deployments of the MicroStrategy platform (specifically, MicroStrategy Web and MicroStrategy Library if available) and runs a user-defined set of end-to-end performance tests on content objects including RSDs (Report Services Documents), reports, Visual Insight dashboards, and Dossiers. The purpose of this tool is to establish a baseline for the performance of a MicroStrategy application or determine how a particular MicroStrategy application’s performance has been affected by an upgrade to a new version by comparing two environments.
Capabilities

- Test the response time of one or many content objects
- Test the response time on one or multiple MicroStrategy environments
- Perform tests by running them once, or many times with saved configurations
- Runs against MicroStrategy 10.4.x and above

Restrictions

- Only non-prompted content objects are supported.
- Connecting to versions of MicroStrategy without MicroStrategy Library will require the user to provide a list of which objects to test.
  Refer to the section *Configuration Files: Profiles.csv* for more information.
- Connecting to MicroStrategy Web 10.4 Hotfix 5 and earlier will require the Wait Page to be disabled.
  Refer to the section *Disabling the Wait Page in MicroStrategy Web* for more information.
- The Capacity Testing Tool only runs in command-line mode
- The Capacity Testing Tool only allows for MicroStrategy users with Standard or LDAP authentication
- The Capacity Testing Tool requires that the user know some configuration properties of their MicroStrategy deployment (see *Gathering MicroStrategy Configurations* for more information).
Getting Started

First, download the zip file from the MicroStrategy Download site and unzip it to a folder. We will refer to this as “Capacity Testing Tool Folder” going forward.

Capacity Testing Tool System Requirements

- Windows Server 2012 or 2016
- 1 GB minimum hard drive storage space
- 8 GB RAM
- Network connectivity to all MicroStrategy Web and MicroStrategy Library (if applicable) instances that will be used

MicroStrategy Requirements

- MicroStrategy Web connected to your Intelligence Server
- MicroStrategy Library 10.9 and above (to use the search functionality)
- A MicroStrategy user that has execute permission for all content objects to be tested
- Non-prompted content objects
- The Web statistics panel must be open on your screen. To enable the Web statistics panel, see Before Running the Test.

Disabling the Wait Page in MicroStrategy Web

The wait page in MicroStrategy Web needs to be manually disabled when connecting to MicroStrategy 10.4 Hotfix 5 and earlier. This is because the URL flag to disable the wait page didn’t exist until MicroStrategy 10.4 Hotfix 6. Before using the Capacity Testing Tool, follow the steps below to disable the wait page:
Disabling the Wait Page on J2EE Environments

1. Log in to the web server and stop the service.

2. Go to the deployment folder (Ex. webapps in Tomcat) where the MicroStrategy.war file was deployed.


5. Open pageConfig.xml in a text editor and do a search for all elements named maxWait and set the value to -1 for all instances.

6. Save the file and exit the editor.

7. Restart the web server service.

Disabling the Wait Page on .NET Environments

1. Log in to the web server and stop the Internet Information Services (IIS) Manager service.

2. Go the folder Web ASPx where web server is deployed. Default path should be C:\Program Files (x86)\MicroStrategy\Web ASPx).

3. Navigate to Web ASPx\WEB-INF\xml and find the file pageConfig.xml.


5. Open pageConfig.xml in a text editor and do a search for all of the elements named maxWait and set the value to -1 for all instances.

6. Save the file and exit the editor.

7. Restart the IIS Manager service.
Once testing has been done, be sure to stop your web environments, and revert back the changes by using the `.bak` files or manually updating the `pageConfig.xml` file.

Gathering MicroStrategy Configurations

For the Capacity Testing Tool to connect successfully you will need to know the configuration properties for the environments to be tested. Most of this information can be collected by logging into MicroStrategy Web or MicroStrategy Library pages and observing the following items in the URL of a web browser.

- **Web Protocol**: Highlighted as 1 in the picture below. It will be either `http` or `https`.

- **Web Server Address**: Highlighted as 2 in the picture below.

- **Web Server Port**: If the port is not default, it will appear as a number between 2 and 3.

- **MicroStrategy Web Path**: Highlighted as 3 in the picture below.

- **Intelligence Server IP or FQDN**: Highlighted as 4 in the picture below.

  ![URL example](https://env-117462.customer.cloud.microstrategy.com/)[MicroStrategy/servlet/mstWeb]evt=3010&Server=ENV-117462UI&USE1

  ![URL example](https://env-117462.customer.cloud.microstrategy.com)[MicroStrategyLibrary]auth/ui/loginPage

  If not visible, you can log into Web Administrator and see which Intelligence Server the Web Server is connected to.

- **MicroStrategy Library Path**: Appears between `.com/` and `/auth` in the picture below.
- **Intelligence Server Authentication Mode**: If you are using corporate credentials to log into MicroStrategy, it is likely that LDAP is being used. If you are using a different login, such as the default one that comes with MicroStrategy Cloud Platform for AWS or MicroStrategy Cloud Platform for Azure, the login mode is likely Standard.

- **MicroStrategy Web Type (.NET or JSP)**: If the MicroStrategy deployment is running on Linux, it is likely JSP, if Windows, it is more likely to be ASP. Ask your MicroStrategy administrator if in doubt.

- **MicroStrategy Username and Password**: The account used must have **Execute** permission in MicroStrategy Web for all content objects being tested.

- **Intelligence Server Port**: By default, it is 34952, but if in doubt, ask your MicroStrategy administrator or check the configuration in MicroStrategy Web Administrator.

**Testing an Environment for the First Time**

**Before Running the Test**

1. Log in to your MicroStrategy Web Administrator page.

2. Open **Diagnostics > Statistics** and set the **Mode** drop-down to **Screen**.

3. Click **Save** to open the Statistics panel:
Leave this panel open for the remainder of the Capacity Testing Tool process.

Executing the Capacity Test

Once MicroStrategy configuration information has been gathered and the Statistics panel is open, navigate to the extracted Capacity Testing Tool folder and double-click on the Capacity Testing Tool.exe file to open the program in a terminal window.

Clicking inside the terminal window while the capacity test is running will cause the test to pause.

If a previously saved response file has been created, a JSON file with instructions for the Capacity Testing Tool to automatically run your test is used. This is useful if you’re doing the test repetitively or automating it with a script. Refer to the section Configuration Files: Silent Response File.

Otherwise, proceed by selecting option 2.
The screen will refresh and begin to configure Environment #1. The first question that will be presented is what type of MicroStrategy Web is running. If your MicroStrategy deployment is running on Linux, it is likely JSP; if Windows, it is more likely to be ASP. Ask your MicroStrategy administrator if in doubt.

This tool can execute one or multiple capacity tests against your Intelligence Environment(s)
It is suggested to start with the environment with the most recent version

Follow the questions below to configure your test

ENVIRONMENT #1
Collecting information about your web server properties...

Select the MicroStrategy Web Server type:
[1] ASP (IIS- Windows)
[2] JSP (Universal)
2

The next question will ask for the platform version for this environment.
Only enter numeric values and periods. For example, for 10.4 Hotfix 8, enter 10.4.8.

If you enter a version below 10.9, you will not be prompted for MicroStrategy Library, which will require you to manually specify which content objects to run.

Following the platform version, select the web protocol MicroStrategy Web is using, whether it is http (unsecure) or https (secure). You can infer this from the URL that is being used when accessing MicroStrategy Web via a browser.

Enter the MicroStrategy Platform Version: 11.0
Select the web protocol:
[1] http
Enter the web port that is being used. Observe that the number \[443\] is present as the default for \texttt{https} that was selected, meaning that if you press Enter, it will auto-populate with that value, which is the default port number for \texttt{https}. If you select \texttt{http}, it will default to port 80. Any values that appear in the prompt that are in brackets are default values, and to select them, press Enter.

Enter the web port \([443]\):

Enter the web server address, which you can get from the URL.

Enter the web server address (ex. env-123.us-east-1.elb.amazonaws.com):

Enter the path to MicroStrategy Web, which you can get from the URL (see the diagram in \textit{Testing an Environment for the First Time})

Enter the web path extension \(/[\text{Microstrategy/servlet/mstrWeb}]\):

Enter the IP address or Fully Qualified Domain Name (FQDN) of the Intelligence Server.

This value is from the perspective of MicroStrategy Web connecting to the Intelligence Server, not from the perspective of a client machine. Even if this IP address or FQDN isn’t accessible from your machine, it will still work. You can see this in the diagram in \textit{Testing an Environment for the First Time}, or by looking at the Intelligence Servers on your MicroStrategy Web Administrator page.

Enter the Intelligence Server address (IP or FQDN):

Enter the port that the Intelligence Server service is running on. By default, it is 34952, but ask your MicroStrategy administrator or check the configuration in MicroStrategy Web Administrator to confirm.

Enter the Intelligence Server port \([34952]\):
Select either Standard or LDAP connection mode for your Intelligence Server to execute content objects.

If you are using your corporate credentials to log into MicroStrategy it is likely that you are using LDAP. If you are using a different login, such as the default one that comes with MicroStrategy Cloud Platform for AWS or MicroStrategy Cloud Platform for Azure, your login mode is likely Standard.

Select the connection mode:
[1] Standard
[2] LDAP
1

Enter the username and password that will execute the content objects for the test.

The message below will be received when a connection has been made.

Enter the username for the environment: mstr
Enter the password for the environment: ***********

Validating connection to Web and the Intelligence environment...
Connection successful!

If you entered a version of MicroStrategy 10.9 and above, you will be asked for your MicroStrategy Library API path. It is typically your web address followed by /MicroStrategyLibrary/api or /Library/api, but if you see something different in your URL bar when logged into MicroStrategy Library, provide that here.

Enter the full Library API URL
(ex.

Validating API connection...
Connection successful!
In order to execute the test, a file called Profiles.csv needs to be created, which has instructions about which content objects to run.

Refer to the section Configuration Files: Profiles.csv for more information.

It is only necessary to provide this file if the first connected environment does not have MicroStrategy Library and you do not have a ResultsSummary.csv from a previously executed Integrity Manager test. Select 1 to do so.

If you have previously executed an Integrity Manager test and want to execute the same content objects, select 2.

Otherwise, select 3 to enter the search mode and be guided through the process of selecting which content objects you would like to test.

Collecting information about your test...

Select the situation that best describes you:
I want to...
[1] upload my own Profiles.csv file
[2] upload my ResultsSummary.csv file from Integrity Manager.
[3] search for Reports, Documents, or Dossiers and create a new Profiles.csv

+----------------------------------------+
MAIN SEARCH MENU:
Use \IN <PROJECT NAME(S)> after the keyword to filter on a specific PROJECT

Type \DONE or \D when you are finished selecting your documents/Reports

Type \VIEW or \V to view your current selection list

** All commands must include the BACKSLASH **
Upgrade Guide

Enter the name (or a keyword) of the object you are looking for:

You can search two different ways:

1. Search object by name.
   This allows the user to search the entire metadata for a keyword matching a content object's name.

2. Browse project folders for objects.
   This allows the user to traverse through folders seeing what objects are available to be added.

Start the test by selecting 1.

SEARCH OPTIONS:

- [1] Search object by name
- [2] Browse project folders for objects
- [3] Exit search

Enter option: 1

Searching by name presents the following options:

- \IN <Project Names(s)>: Adding this option to the end of the keyword will search for the keyword in the specified project(s) only. Multiple project names must be comma separated.

- \VIEW or \V: This option allows you to see what objects have been selected so far. Currently the client does not allow the removal of objects.
from the selection. You can modify selections by editing the response file generated at the end of the session and re-run the test using the response file.

- \DONE or \D: Will end the searching session.

Search for the word *supply* to see an example of the results returned.

Enter the name (or a keyword) of the object you are looking for: supply
Starting search for objects with name containing 'supply'... ...
...

SEARCH RESULTS:

+-----------------------------+
<table>
<thead>
<tr>
<th>Project Name, Type, Item Name</th>
</tr>
</thead>
</table>

+-----------------------------+
| Use COMMA (,) to separate the indexes |
| Use DASH (-) to capture a range of indexes |
| Use STAR (*) to select ALL |
| Press ENTER to go back |
+-----------------------------+

Enter the index(es) of the document(s)/report(s) you wish to select:
This is the page where the user searches for an object. The searching tool prints out search results in the following manner:

- Sorted by Project Name, Object Type, Object Name
- There are two spaces between different projects
- If objects with the same name are found in the same project, the result menu will also print out the object’s full path so you can distinguish between them.

Each set of results have an index assigned to it (an integer). You select objects using the index and the following modifiers:

- Select individual indices with commas (,)
- Select indices with a range (-)
- Select all returned results with a star (*)
- User can choose to not select any results and return to main menu by pressing ENTER

For example, if you only wanted to add “1- Supply Chain Management Report” and “Supply Chain Management Report”, then simply enter 1,3.

```
Enter the Index(es) of the document(s)/report(s) you wish to select: 1,3
Input(s) saved! You can see your saved objects by typing \VIEW in main search menu
```

Once you have added all the content objects searching by name, type \D to exit and select 2 to search by traversing through folders.

```
+----------------------------------+
SEARCH BY NAME

Use \IN <PROJECT NAME(S)>
after the keyword to filter on a specific PROJECT
```
Type \DONE or \D
  when you are finished selecting your documents/Reports

Type \VIEW or \V
  to view your current selection list

** All commands must include the BACKSLASH **

Enter the name (or a keyword) of the object you are looking for: \d

SEARCH OPTIONS:

[ 1 ] Search object by name

[ 2 ] Browse project folders for objects

[ 3 ] Exit search

Enter option: 2

PROJECTS:

[ 0 ] Consolidated Education Project
[ 1 ] Hierarchies Project
[ 2 ] Human Resources Analysis Module
[ 4 ] Platform Analytics
From here, users can use the index values to first select a project, and once inside the project, the project name and parent folders will appear. This allows the user to continue down and find the folder that contains the objects they want to add. For example, when trying to travel to the folder /MicroStrategy Tutorial/Public Objects/Reports/Sample Dossiers, here are the values that would be selected to get there.

+---------------------------------------------------------------------+
Type \VIEW or \V
   to view your current selection list
Type \DONE or \D
   when you are finished selecting your
   reports/documents/dossiers

** All commands must include the BACKSLASH **
+---------------------------------------------------------------------+
Select a project to start drilling down folders: 3

+---------------------------------------------------------------------+
FOLDERS:

/MicroStrategy Tutorial
[ 0 ] Project Builder
[ 1 ] Project Objects
[ 2 ] Public Objects
[ 3 ] Schema Objects
TO SELECT OBJECTS...

Use COMMA (,) to separate the indexes
Use DASH (-) to capture a range of indexes
Use STAR (*) to select ALL
Press ENTER to go back

Enter the index(es) of the object(s) you wish to select OR drill further into a folder: 2

FOLDERS:

/MicroStrategy Tutorial/Public Objects
[ 0 ] AutoStyles
[ 1 ] Consolidations
[ 2 ] Custom Groups
[ 3 ] Documents
[ 5 ] Filters
[ 7 ] Prompts
[ 8 ] Reports
[ 9 ] Searches
[ 10 ] Templates

+--------------------------------------------------------------------------+

TO SELECT OBJECTS...

Use COMMA (,) to separate the indexes
Use DASH (-) to capture a range of indexes
Use STAR (*) to select ALL
Press ENTER to go back

+--------------------------------------------------------------------------+

Enter the index(es) of the object(s) you wish to select OR drill further into a folder: 8

+--------------------------------------------------------------------------+

OBJECTS:
Project Name, Type, Item Name
[0] MicroStrategy Tutorial, Report, I am a Report

FOLDERS:
/MicroStrategy Tutorial/Public Objects/Reports
[ 2 ] Business Roles
[ 3 ] Documents and Scorecards
[ 4 ] Enterprise Reporting Documents
[ 7 ] Sample Dossiers
[ 8 ] Subject Areas

TO SELECT OBJECTS...
Use COMMA (,) to separate the indexes
Use DASH (-) to capture a range of indexes
Use STAR (*) to select ALL
Press ENTER to go back

Enter the index(es) of the object(s) you wish to select OR drill further into a folder: 7
OBJECTS:

Project Name, Type, Item Name

[0] MicroStrategy Tutorial, Dossier, Advanced and Predictive Analytics


+FOLDERS:

Upgrade Guide
You can see there are 12 dossiers in this folder with the indexes 1 through 11. The same options to add objects to your test apply here as they did above with a common, dash, and star options. Once the objects are selected from the folder, the search will return to the project level to start again.

Once all of your objects have been added to your test, use \V to quickly see what has been added before selecting \D to exit the folder search, followed by 3 to close the search functionality.
[ 1 ] Hierarchies Project
[ 2 ] Human Resources Analysis Module
[ 4 ] Platform Analytics

+---------------------------------------------------------------+
-----------------------------+
Type \VIEW or \V
       to view your current selection list
Type \DONE or \D
       when you are finished selecting your
       reports/documents/dossiers

** All commands must include the BACKSLASH **
+---------------------------------------------------------------+
-----------------------------+
Select a project to start drilling down folders: \v

+-----------------------------+

YOUR SELECTION:

+-----------------------------+
### Project Name, Type, Item Name

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>MicroStrategy Tutorial, Document, Supply Chain Management</td>
</tr>
<tr>
<td>2</td>
<td>MicroStrategy Tutorial, Dossier, Advanced and Predictive Analytics</td>
</tr>
<tr>
<td>3</td>
<td>MicroStrategy Tutorial, Dossier, Custom Visualizations</td>
</tr>
<tr>
<td>4</td>
<td>MicroStrategy Tutorial, Dossier, Finance Executives</td>
</tr>
<tr>
<td>5</td>
<td>MicroStrategy Tutorial, Dossier, Geospatial Features</td>
</tr>
<tr>
<td>6</td>
<td>MicroStrategy Tutorial, Dossier, Investment Firm Dossier</td>
</tr>
<tr>
<td>7</td>
<td>MicroStrategy Tutorial, Dossier, Life as a New Hire</td>
</tr>
<tr>
<td>9</td>
<td>MicroStrategy Tutorial, Dossier, Multinational Bank Dossier</td>
</tr>
<tr>
<td>10</td>
<td>MicroStrategy Tutorial, Dossier, Office Royale Sales</td>
</tr>
<tr>
<td>11</td>
<td>MicroStrategy Tutorial, Dossier, Retail Store Manager</td>
</tr>
<tr>
<td>12</td>
<td>MicroStrategy Tutorial, Dossier, Slot Machine Performance</td>
</tr>
</tbody>
</table>

+-----------------------------------------------+

PROJECTS:

[ 0 ] Consolidated Education Project
[ 1 ] Hierarchies Project
[ 2 ] Human Resources Analysis Module
[ 4 ] Platform Analytics

+-----------------------------------------------+

Type \VIEW or \V
to view your current selection list
Type \DONE or \D
when you are finished selecting your
reports/documents/dossiers

** All commands must include the BACKSLASH **

+-----------------------------------------------+

Select a project to start drilling down folders: \d

+-----------------------------------------------+
SEARCH OPTIONS:

[ 1 ] Search object by name
[ 2 ] Browse project folders for objects
[ 3 ] Exit Search

Enter option: 3

Now the test execution configuration needs to be set, by setting how many jobs (threads) to execute simultaneously against the Intelligence Server, and how many times (cycles) to perform each set of steps.

One cycle would include logging in as the defined user, executing all of the content objects, and then logging out.

Enter y to accept the default test configuration or n to define a custom test.

Collecting properties about your test...

Here are the default threads and cycles:
1. Thread: 1  Cycle: 5
2. Thread: 1  Cycle: 5
3. Thread: 2  Cycle: 5
4. Thread: 3  Cycle: 5
5. Thread: 4  Cycle: 5
6. Thread: 5  Cycle: 5

Would you like to keep the default capacity testing properties? (y/n):

Below is an example of defining a custom test.

First, define the number of simultaneous jobs that will be executed against the Intelligence Server. Second, define the number of times that the test set will be executed. After the first execution is defined, you may continue adding additional executions by entering y or n to move forward.
TEST CYCLE #1
Enter the number of thread(s): 2
Enter the number of cycle(s): 4
Would you like to add another JMeter test cycle? (y/n): n

Once the test configuration has been completed, another environment can be selected. For example, the test above was configured with Environment #1 on 2019, and if the user is upgrading from 10.4.8, for the second environment they will want to input that connectivity information. As many as 10 different environments can be defined within the same test to collect capacity information from those environments simultaneously.

Would you like to add another environment test? (y/n): y

ENVIRONMENT #2
Collecting information about your web server properties...

Select the MicroStrategy Web Server type:
[1] ASP (IIS-Windows)
[2] JSP (Universal)
2
Enter the MicroStrategy Platform Version: 10.4.8
Select the web protocol:
[1] http
[2] https
2
Enter the web port [443]:
Enter the web server address (ex. env-123.us-east-1.elb.amazonaws.com): env-113986.customer.cloud.microstrategy.comEnter the web path extension [/MicroStrategy/servlet/mstrWeb]: Enter the Intelligence Server address (IP or FQDN): 10.250.144.255 Enter the Intelligence Server port [34952]:

Select the connection mode:
Enter the username for the environment: mstr
Enter the password for the environment: ************

Validating connection to the environment...
Connection successful!

Would you like to add another environment to test (y/n): n

Once all environments have been added and the user selects n, the test configuration will be complete and a JSON response file will be automatically created in the same directory where the Capacity Testing Tool exists. See Testing an Environment for the First Time for instructions on how to re-run your test later.

Select 1 to execute the saved test against the environments defined, or select 2 to close the Capacity Testing Tool, which will close the program.

If there are any problems that occur during the test, jmeter.log will be created under apache-jmeter-3.1\bin\ and additionally under the folder Capacity_Results_Analysis for the test’s execution results.
Testing an Environment Again

Configuration Files: Profiles.csv

In order to execute tests, the Profiles.csv file containing the list of content objects to run, is required. It is only necessary to provide this as a standalone file if the first environment does not have MicroStrategy Library. Otherwise users can select 2 and be guided through the process of selecting content objects. However, if 1 is selected, or a user does not have MicroStrategy Library configured, Profiles.csv must be provided.

The CSV file specifies the objects that will be used for the capacity test. As stated above, the Capacity Testing Tool supports reports, Report Services Documents, dashboards, and dossiers. Prompted objects of any kind are NOT supported. The file is populated with sample data by default to help illustrate the format needed.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroStrategy Tutorial</td>
<td>Administrator</td>
<td>2alkdf</td>
<td>B -- Make Recommendation</td>
<td>reportID</td>
<td>1573B66C42306028A8941E080B918CEAE4C</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Tutorial</td>
<td>Administrator</td>
<td>2alkdf</td>
<td>The 2014 BNY Mellon Boat Race</td>
<td>reportID</td>
<td>265485A411E41B5E000000B000F7F7B2FF</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Tutorial</td>
<td>Joshua</td>
<td>happy89</td>
<td>Category Sales &amp; Profitability Dashboard</td>
<td>documentID</td>
<td>E05D4B3421DB3C5FCE9B4D123AE9D</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Tutorial</td>
<td>Clayton</td>
<td>30003Met</td>
<td>Product Performance</td>
<td>documentID</td>
<td>B753F3B48FC5C1764061896512F05</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Tutorial</td>
<td>Administrator</td>
<td>2alkdf</td>
<td>Corporate Sales Overview</td>
<td>documentID</td>
<td>E080D66347A0DD914D3B568512F05</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Tutorial</td>
<td>Executive</td>
<td>oia409</td>
<td>Operational Performance Dashboard</td>
<td>documentID</td>
<td>BF2ACEDE413F25131B6F690931B26</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Tutorial</td>
<td>Administrator</td>
<td>2alkdf</td>
<td>Predictive Customer Scorecard</td>
<td>documentID</td>
<td>374DAB7E11679620AE4B085F5AF736</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Tutorial</td>
<td>Administrator</td>
<td>2alkdf</td>
<td>Category Sales Report</td>
<td>documentID</td>
<td>0CC4A711E679F061920D0EF151179</td>
<td></td>
</tr>
<tr>
<td>MicroStrategy Tutorial</td>
<td>Administrator</td>
<td>2alkdf</td>
<td>Product Analysis Report Builder</td>
<td>reportID</td>
<td>D529D3421E679F061920D0EF151179</td>
<td></td>
</tr>
</tbody>
</table>

There are six columns that need to be populated in the CSV file for each object to be correctly executed:
- **Project Name**: This needs to be exactly as it is displayed in Web.

![Project Name](image)

- **User login**: This is the login for the user that will execute the object with. Keep in mind that you can use any combination of users for the test. In other words, exact object can be executed with the same login, or a combination. The results will inherit the object security and security filters as if they were logging into web, so be sure that you pick appropriate users.

- **User password**: Password for user’s login. If the password is blank, leave it empty.

- **Name of object to be executed**: This is the name of the object that will be displayed in the capacity test results. While it is recommended that the name used should be exactly how it exists in the project, keep in mind that special characters are not supported, and therefore remove them here. You do not need to worry about renaming the actual objects in each project.

- **Object type**: If the object is a report, use “reportID” for this field. If the object is a Report Services Documents, dashboard, or dossier use “documentID.”

- **Object ID**: The corresponding ID of the object to be tested. In Web, right-click on the object’s name, and click Properties which will provide you the ID that is needed.
Configuration Files: Silent Response File

You can provide a JSON file with instructions for the Capacity Testing Tool to automatically run a test. This is useful if you’re doing the rest repetitively or automating it with a script. To use a silent response file, select option 1 for the first prompt that is provided.

The tool generates a response file after a test is created. This response file is a JSON file that can be used again by the client to re-run a test. The response file by default will be placed under the capacity test package and will be named with the format `response_full_environment_hostnames_timestamp.json`. The user can edit this at their own discretion.

```
#Welcome to the #
#MicroStrategy Performance & #
#Capacity Test Tool #

[1] Run test using response file
[2] Start a new test
```

An example of a response file is as follows.

```
{
   "Tests": [
   
   "WebServerProperties": {
       "webServerType": "JSP",
       "webServerAddress": "env-1234.customer.cloud.microstrategy.com",
       "webPort": 443,
       "protocol": "https",
       "webPath": "/MicroStrategy/servlet/mstrWeb",
       "iServerName": "env-1234laiouse1",
       "iServerPort": 33495,
       "connMode": 1,
       "mstrVersion": "11.0"
    },
   
   }]
```
"Profiles": [
  {
    "projectName": "MicroStrategy Tutorial",
    "username": "mstr",
    "password": "password123",
    "name": "Finance",
    "type": "report",
    "ID": "RE12344GGDI1123444",
    "path": "/MicroStrategy Tutorial/Sample Dossier/Finance/"
  },
  {
    "projectName": "",
    "username": "",
    "password": "",
    "name": "",
    "type": "",
    "ID": "",
    "path": ""
  }
],
"JMeterProperties": [
  {
    "thread": 1,
    "cycle": 2
  },
  {
    "thread": 1,
    "cycle": 5
  },
  {
    "thread": 5,
    "cycle": 5
  }
]
Troubleshooting the Capacity Test

This section contains common errors encountered while executing a Capacity Test and how to resolve those errors.

Capacity Testing tool times out a report execution

If larger reports or documents are not fully executed during the test you will need to increase the timeout limit.

How to Increase the Timeout Limit

1. In the extracted Capacity Testing Tool folder, open jmeter > templates > template_while.

2. In line 17, modify the parameter `parseInt(${counter}) != 300` to a number between 100 and 400.

   For example,

   ```
   WhileController.condition">=="${_javaScript( "${DONE}" != "mstr-stats" &&
   parseInt(${counter}) != 100 ;})"</stringProp>
   ```

Capacity Testing Tool pauses during testing

- Capacity Testing tool will pause the test if you click anywhere on the terminal (black screen). If you click on the terminal, to resume the test, hit Enter on your keyboard.

[HTTPError] 401 Unauthorized

Possible Solutions:

- Double check that the password was entered correctly.

- Ensure that the URL entered is correct.

- IIS users should ensure that Anonymous Authentication is enabled and Windows Authentication is disabled.
You can check this by pasting the MicroStrategy Web URL into a browser running in Private or Incognito mode. If you are prompted for credentials, check the above authentication settings.

- Check that the account being used for the Capacity Test has permissions for the MicroStrategy folder on the Web Server.

[HTTPError] 302

- Check that the account being used for the Capacity Test has permissions for the MicroStrategy folder on the Web Server.

If your Profiles.csv file is in a language other than English

- Open this csv file with Notepad++ to ensure that there are no erroneous characters in the file. These text encoding issues will cause the Capacity Testing Tool to report errors.

Test failed: text expected to contain /Shared Reports/

- This error may appear when your Web language settings are set to a language other than English. To resolve the error, set the Web language to English.

Test failed: text expected to contain /pageWait.css/

- This error may appear when running a Capacity Test on environments using MicroStrategy version 10.4 through 10.4.5 or version 10.11. Refer to Disabling the Wait Page in MicroStrategy Web to resolve this issue.

Upgrade Analysis Dossier

The Upgrade Analysis Dossier consumes the results of comparison testing in MicroStrategy Integrity Manager and the new MicroStrategy Capacity Testing Tool to help Platform Administrators understand the impact of upgrading from engine changes along with performance improvements.

The most recent dossier can be downloaded here.
This section will explain how to upgrade the dossier with new results from both Integrity Manager and the Capacity Testing Tool, along with explaining the use of each Chapter and its corresponding Pages.

**Refreshing Upgrade Impact Results**

1. Navigate to the output folder that was created after comparison testing was done in Integrity Manager.

   There are five files generated after running *Baseline vs Project* in Integrity Manager:
   - `BaseObjectDependency.csv`
   - `BaseUpgradeImpact.csv`
   - `ResultsSummary.csv`
   - `TargetObjectDependency.csv`
   - `TargetUpgradeImpact.csv`

2. Copy the five files to a client machine where Workstation is installed. Once there, open the Upgrade Analysis Dossier.

3. Enable the **Datasets Panel** and click the three dots next to `ResultsSummary.csv` and select **Edit Dataset**.

4. Click on the down-arrow next to the `ResultsSummary.csv` table and select **Edit Table**.

5. When the screen opens, click on **Choose files** and then select your `ResultsSummary.csv` file and then click **Open > Refresh**.

6. Repeat steps 3 - 5 with the following Tables > Files:
   - **BaseObjectDependency > BaseObjectDependency.csv**
   - **TargetObjectDependency > TargetObjectDependency.csv**
7. Once all of the files have been refreshed, click **Update Dataset** to update the entire dataset.

### Refreshing Capacity Results

1. Navigate to the output folder that was created after comparison testing was done in Integrity Manager.

   Within that folder identify two CSV files:
   
   - **Execution_Details.csv**
   - **Summary.csv**

2. Copy the two files to a client machine where Workstation is installed. Once there, open the Upgrade Analysis Dossier.

3. Enable the **Datasets Panel** and click the three dots next to **Execution_Details.csv** and select **Edit Dataset**.

4. Click on the down-arrow next to the **Execution_Details.csv** table and select **Edit Table**.

5. When the screen opens, click on **Choose files** and then select your **Execution_Details.csv** file and then click **Open > Refresh**.

6. Repeat steps 3 - 5 for the **Summary.csv** table.

7. Once all of the files have been refreshed, click **Update Dataset** to update the entire dataset.

### Examining Testing Results

Once the two datasets have been refreshed, the dossier is now ready to be examined. Keep in mind before getting started that unless the same environments and content objects (reports, Report Services Documents, and dossiers) were used for the Integrity Manager comparison testing and
for Capacity Testing Tool, then the results that appear in the two chapters explained below will be independent of each other.

Upgrade Impact Results Chapter

Comparison testing with MicroStrategy Integrity Manager now provides logging that helps provide a full analysis of differences found when comparing MicroStrategy 2020 environments to MicroStrategy 10.x environments with an explanation of the cause and the reasons for such changes. This chapter of the Upgrade Analysis Dossier provides four different pages illustrating those results. These pages help to identify object mismatches that are affected by version changes. More importantly, objects that have a mismatch but did not encounter a change and content objects that should examined in more detail, are now easily identifiable.

Object Overview Page

The Object Overview page shows the same summary information that is contained in the ResultsSummary.html file. At the top, Platform Administrators can see the full build version from the Intelligence Servers that were tested.

The Object Overview master grid provides all of the objects that were tested along with their comparison status and performance improvement if multiple executions were done.
A simple guide to understanding the results:

<table>
<thead>
<tr>
<th>Result</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>◼</td>
<td>Executions successful or comparison matched between Base and Target</td>
</tr>
<tr>
<td>♦</td>
<td>Comparison mismatched between Base and Target</td>
</tr>
<tr>
<td>Not Tested</td>
<td>Option not selected to be tested</td>
</tr>
<tr>
<td>Comparison Unavailable</td>
<td>Comparison is not supported for object type</td>
</tr>
</tbody>
</table>

The master grid is used as a selector for the bottom grids **Object Path & Changes** and **Dependent Objects & Changes**. When you select an object on the master grid, that object will target and filter the two bottom grids. If the object encountered a change, then the Object Path & Changes grid will list the object’s name, path, the change identifier (Ex. US120878) and a “link.” This link when clicked will open up the default browser on the
machine and navigate to a MicroStrategy Knowledge Base article to help explain any mismatches that the object encountered. The “Dependent Objects & Changes" is used to identify any dependent objects that were executed for the object that was selected that also encountered any changes.

For example, an object called Document1 used three datasets Report1, Report2, and Report3. When you select Document1 from the master grid, the “Object Path & Changes" appears as blank, but Report1 and Report2 appear in the “Dependent Objects & Changes" grid, both showing with the change identifier DE123456. This means that when Report1 and Report2 were executed against the Intelligence Server to gather the dataset results for Document1, they encountered a change, however when Document1 was being processed, it did not encounter a change. This helps to identify the specific object where a change was encountered, whether it is at the parent object level, or dependent object level.

Changes Encountered Page

The Changes Encountered page quickly provides Platform Administrators with the number of changes that were returned from the execution of all content objects and their dependent objects. With the “Changes" grid at the top, select any change identifier (first column) or its corresponding description to filter the bottom grid “Objects Affected" to see exactly which objects from the test were affected by the change.
Keep in mind that as new changes continue to get added, Knowledge Base articles are being written thereafter so some may not have links yet, but they will available soon.

Object Mismatches without Changes Page

The Object Mismatches w/o Changes page lists those objects that were tested and returned a comparison mismatch for SQL Comparison or Data Comparison but did not encounter any changes that can explain the mismatch. Although these mismatches could be due to explainable things including different environments pointing to different warehouses, dynamic date prompts, etc., they should be examined further with Integrity Manager’s ResultsSummary.html to see the exact SQL and Data differences to help explain why they are not matching.
Performance Evaluation Page

If performance testing was done within Integrity Manager by running multiple executions of content objects, then this bubble graph will help quickly identify if their performance was better or worse when comparing the Base versus Target versions.

The bottom grid “Object Performance Cycles” will show the average performance gain or loss of objects by type. To see the specific gain or loss for a content object, select a bubble to filter the bottom grid.
Capacity Testing Results

The MicroStrategy Capacity Testing Tool provides users with a quick and easy way to perform end-to-end performance testing. This chapter provides three different pages illustrating those results, helping to identify any throughput or resource problems, as well as comparing different platform releases to confirm performance gains, and identify any problem areas, down to the step-level.

Summary Page

The Summary page provides with you a high-level overview of the projects tested by users across versions to provide an average response time comparison for quick analysis. The bubble chart below shows the average response for each version in respect to the number of threads, or jobs, that were executing simultaneously from the tool.
As the number of threads increases and stress on the Intelligence Server increases, errors will start to be received which increases the size of the bubbles, illustrating stress on the server. This helps the Platform Administrator when the server is starting to hit the maximum capacity that it can handle based on resources and the tuning that has been done.

Performance by Execution Type Page

The end-to-end testing that the tool provides includes logging in and logging out as part of the execution steps. With the Performance by Execution Type page, Platform Administrators can see the performance of only the content objects or the performance of logging in and logging out only.
Execution Details Page

The Execution Details page uses an outline grid with the same name at the top to provide every single execution value for every step down to the thread used. Clicking on a step will target and filter the bottom grid Execution Step Comparison by Version. As with the Summary page graph, Platform Administrators can see the performance for each thread for an exact step for a specific user.

This view can help in identifying outliers or nuances in performance from one step for a content object or logging in/logging out that may be performing poorly.
Resolving Problems Encountered During the Upgrade

The Readme contains troubleshooting scenarios aimed to assist you during or after an upgrade to the latest version of MicroStrategy. If you encounter an issue not covered in the Readme or by the suggestions below, contact MicroStrategy Technical Support for additional assistance.

- Make sure you have met the system requirements and other prerequisites for the current version of MicroStrategy, as noted in Upgrade Prerequisites, page 7.

- Be aware of the level of interoperability between the current version of MicroStrategy and the version you are upgrading from, as noted in Compatibility and Interoperability section of the Readme.
- Any errors in the upgrade process are likely to be written to the installation log file, `install.log`. This file is in the directory that you specify as the installation directory during the upgrade.

- When you use the Configuration Wizard to add tables to the metadata or update your projects, each SQL statement that is executed is logged to the Configuration Wizard log file, `MACfgWiz.xml`. This log file is in the Log subfolder of the main MicroStrategy directory. You can analyze the log file using a third-party log viewer, such as the Apache Chainsaw log viewer, which can be downloaded from [http://logging.apache.org/chainsaw/index.html](http://logging.apache.org/chainsaw/index.html).

- If MicroStrategy Web users encounter errors after you upgrade MicroStrategy Web, you may need to clear your application and web server caches, or your users may need to clear their browser caches. For instructions on how to clear these caches, see the documentation for your web server, application server, and web browser.

- MicroStrategy recommends upgrading to the latest version without uninstalling the previous version of MicroStrategy. However, in rare situations, issues may be encountered as a result of an in-place upgrade. In these situations, you may need to uninstall your existing MicroStrategy installation before installing the new version.
Supplemental Information

Application Servers

The sections below provide instructions for setting up different application servers to deploy MicroStrategy Web, Mobile, and Library.

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.

You must have administrative privileges to deploy MicroStrategy Web for your project. If this is the first time you are logging in and you have not changed the default MicroStrategy administrative login, you can use Administrator as the login with no password. After the first time, the user name and password should be changed for security purposes.

- The Microsoft Windows' Users group must have read and execute permissions to all of the files within the MicroStrategy common files folder. This ensures that Internet Information Services has the required permissions to host MicroStrategy Web. By default, this folder is stored in the following directory location:

  64-bit Windows environments:
To connect MicroStrategy Web to your Intelligence Server


2. Type the name of your Intelligence Server in the Add a server manually box on the MicroStrategy Web Administrator page.

3. Click Connect. All projects loaded on the Intelligence Server are now available from MicroStrategy Web. Click the Home icon to see the list of projects loaded on the Intelligence Server you specified.

4. Send your users the URL:

   http://webservername/MicroStrategy/asp/

   where webservername is the name of the computer hosting your Web server. For example, if the name of your Web server machine is Web_Srv1, then the URL your users would use to access MicroStrategy Web would be

   http://Web_Srv1/MicroStrategy/asp

You have manually connected MicroStrategy Web to the Intelligence Server.

You can also connect automatically whenever MicroStrategy Web Server or Intelligence Server starts.
To make MicroStrategy Web connect to the Intelligence Server automatically


2. Select the **Automatically connect to Intelligence Server when Web Server or Intelligence Server is restarted** option and click **Save**.

Deploying Mobile Server

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

! You must have administrative privileges to deploy MicroStrategy Mobile Server for your project. If this is the first time you are logging in and you have not changed the default MicroStrategy administrative login, you can use **Administrator** as the login with no password. After the first time, the user name and password should be changed for security purposes.

- The Users group for Microsoft Windows must have read and execute permissions to all of the files within the MicroStrategy common files folder. This ensures that IIS has the required permissions to host MicroStrategy Mobile Server. By default, this folder is stored in the following directory location:

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

To connect MicroStrategy Mobile Server to your Intelligence Server

1. In Windows, go to **Start > Programs > MicroStrategy Tools > Mobile Administrator**.
2. Type the name of your Intelligence Server in the **Add a server manually** box on the MicroStrategy Mobile Server Administrator page.

3. Click **Connect**.

4. Click **Mobile Configuration** to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the **MicroStrategy Mobile Administration Guide**.

You have manually connected MicroStrategy Mobile Server to the Intelligence Server.

You can also connect automatically whenever MicroStrategy Mobile Server or Intelligence Server starts.

---

**To make MicroStrategy Mobile Server connect to the Intelligence Server automatically**


2. Select the **Automatically connect to Intelligence Server when Mobile Server or Intelligence Server is restarted** option.

3. Click **Save**.

---

**Deploying 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 217**: configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP).

Preconfiguration Information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on JBoss on your machine.

Installing the JDK

If you have not installed Oracle JDK yet, download it [here](#). Be sure to install the JDK and not the JRE software options.

To configure the JDK, a system variable must point to the folder where you install the JDK. If you install the JDK to a simple folder path such as `C:\`, then setting the system variable is easier and more likely to be correct.

Configuring the JDK

1. On your Windows machine, go to **Start > Computer > System properties > Advanced system settings > Environment Variables**.

   The third-party products discussed below are manufactured by vendors independent of MicroStrategy, and the steps to configure these products is subject to change. Refer to the appropriate Microsoft documentation for steps to access and modify the environment variables.

2. Under **System Variables**, click **New** to create a system variable. The New System Variable dialog box opens.

3. In the Variable Name box, type **JAVA_HOME**.

4. In the Variable Value box, specify the destination folder where you installed the JDK and click **OK**.
For example, if the fully qualified path to your JDK executable is C:\jsdk1.8.0\bin\java.exe, the value of your JAVA_HOME variable is C:\jsdk1.8.0.

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\jsdk1.8.0.

Installing JBoss

You can download and install JBoss here.

Keep track of the location in which you install JBoss, as this location is used later (referred to as JBOSS_HOME) to configure JBoss with a MicroStrategy JSP application deployment.

Deploying MicroStrategy Web and Mobile Server

Assuming you have made all the necessary configurations described above, you can begin deploying MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) with JBoss.

Deploying using JBoss as a stand-alone Web container

1. Locate the WAR file for your MicroStrategy JSP application.

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 221.*

---

**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 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

**Deploying with Oracle 10g (Windows)**

This chapter provides information used to deploy and configure MicroStrategy JSP applications with Apache as the Web server and Oracle Application Server 10g R3 as the application server. You can use the procedure below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP): For detailed deployment instructions, please see: *Deploying MicroStrategy Web and Mobile Server, page 222*

**Deploying MicroStrategy Web and Mobile Server**

After your machine is configured, you can start the deployment of your MicroStrategy JSP application with Oracle Application Server 10g R3.
Deploying using the Oracle Enterprise Manager

You can access Oracle Enterprise Manager from the following URL:

http://MachineName:PortNumber/em

Where MachineName is the machine name or IP address of the Oracle machine, and PortNumber is the port number of Oracle Enterprise Manager.

1. Start the Apache Web Server. From the Start menu, go to OracleAS 10g - DEFAULT_HOME1 > Start ApplicationServerName.MachineName.domain.
2. To verify that the Apache Web Server has started, open Oracle Enterprise Manager, select HTTP Server, and then click Start.
3. Select the OC4J instance where you want to deploy your MicroStrategy JSP application. This procedure assumes you are using the default instance name home. Click home. The OC4J: home page opens.
4. Select the Applications tab.
5. Click Deploy.
6. In the Archive area, select Archive is present on local host.
7. Click Browse to navigate to and select the WAR file for your MicroStrategy JSP application.
8. In the Deployment Plan area, select Automatically create a new deployment plan and click Next.
9. Enter the Application Name and Context Root. This section on deploying MicroStrategy Web (JSP) with Oracle 10g uses MicroStrategy as the Application Name and /MicroStrategy as the Context Root. For Mobile Server (JSP), this section uses MicroStrategyMobile as the Application Name and
/MicroStrategyMobile as the Context Root.

10. Click Next.

To map a user to the admin security role

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must assign users the security role of admin. In Oracle 10g, the security users and groups are defined in the Oracle Enterprise Manager.

1. In the Map Security Roles task name, click the Go To Task (pencil) icon.

2. For the admin security role, select the Map Role (pencil) icon.

3. Select Map selected users and groups to this role.

4. In the Map Role to Users area, in the User field, type the user name to map to the admin security role and click Add.

   Repeat this step to add all users for whom you want to grant permission to work in the MicroStrategy Web Administrator and Mobile Server Administrator pages.

5. Click Continue and OK.

6. Click Deploy.

7. Stop and restart the Apache Web Server.

Now you can access and configure your MicroStrategy JSP application, as described in Accessing the MicroStrategy JSP administrative pages, page 224.

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 223 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 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 SAP NetWeaver (Windows)

This section provides information used to deploy and configure MicroStrategy JSP applications on a Windows machine using the SAP application server. You can use the procedure below to deploy MicroStrategy Web (JSP) and MicroStrategy Mobile Server (JSP):

For detailed deployment instructions, please see: Deploying MicroStrategy Web and Mobile Server, page 226

Deploying MicroStrategy Web and Mobile Server

Once your machine has the necessary settings configured, you can deploy MicroStrategy Web (JSP), Mobile Server (JSP) (JSP) on the SAP-Windows machine.

Deploying MicroStrategy JSP applications with the SAP NetWeaver Application Server

Follow the steps provided in this section to deploy MicroStrategy JSP applications as a WAR file.
To deploy MicroStrategy JSP applications as a WAR file

1. Locate the WAR file for your MicroStrategy JSP application.

2. Copy the WAR file to the Windows machine hosting your application server. The location in which you store the file is used later and referred to as path_to_war_file.

3. From the Windows Start menu, select Run.

4. In the Open drop-down list, type cmd, and click OK.

5. Using the command prompt, browse to the following directory within the SAP Application Server installation directory:

/usr/sap/SID/Instance_Number/j2ee/deployment/scripts/

The SID and Instance_Number parameters are defined during installation and configuration of the SAP Application Server. The port number above refers to the P4 port number. The default port number is 50004.

6. Type the following command and press Enter to deploy the WAR file:

   Deploy.bat
   user_name:password@localhost:port_number path_to_war_file

   The user name and password must have administrative access. The port number above refers to the P4 port number. The default port number is 50004.

7. Access NetWeaver web admin console using the following URL:

   http://localhost:PortNumber/nwa

   The PortNumber above refers to the J2EE engine port number. The default port number is 50000.

8. Log in as an administrative user.

9. Go to Operation Management > Systems > Start & Stop.
10. Select **Java EE Applications**.

11. Select the MicroStrategy JSP application just deployed from the list.

12. Go to **Application Details > Status > Start**.

13. Select **On all instances and Set "Started" as Initial State**.

Configuring administrative access to MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must map users or groups to the admin security role. This security role is defined in the MicroStrategy JSP application deployment, within the `web-j2ee-engine.xml` file. You can modify this file to map users or groups to this admin security role, or include users in the administrators user group.

Accessing the MicroStrategy JSP applications

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

You must have administrative privileges to access the MicroStrategy Web Administrator or Mobile Server Administrator page. For more information, see Configuring administrative access to MicroStrategy JSP applications, page 228.

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`
For Mobile Server (JSP):

http://MachineName:
PortNumber/
/MicroStrategyMobile/servlet/mstrWebAdmin

The servlet names at the end of the URLs listed above are case-sensitive. Use the correct case when typing the servlet name.

The login dialog box opens.

2. Specify a user name and password.

3. After you are authenticated:

- If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.

- If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click **Mobile Configuration** to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the **MicroStrategy Mobile Administration Guide**. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

4. If you are deploying MicroStrategy Web (JSP), access the MicroStrategy Web Application on SAP Web Server by specifying the following URL in the Web browser:

http://MachineName:PortNumber/MicroStrategy/servlet/mstrWebAdmin
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 230**: Configuration that must occur before you begin deploying MicroStrategy Web (JSP), Mobile Server (JSP).


Preconfiguration Information

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications on Tomcat on your machine.

Installing the JDK

If you have not installed the Oracle JDK yet, download the file from the website. Be sure to install the JDK and not the JRE software options.

To configure the JDK, a system variable must point to the folder where you install the JDK. If you install the JDK to a simple folder path such as C: \ setting the system variable is easier and more likely to be correct.

Configuring the JDK

The third-party products discussed below are manufactured by vendors independent of MicroStrategy, and the steps to configure these products is subject to change. Refer to the appropriate Microsoft documentation for steps to access and modify the environment variables.
1. From the **Start** menu, go to **Computer** > **System properties** > **Advanced system settings** > **Environment Variables** > **System Variables**.

2. Under , click **New** to create a system variable.

3. In the **Variable Name** field, type **JAVA_HOME**.

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

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

   If you have installed JDK under the **Program Files** folder, type 
   **Progra~1** when specifying the folder name in the Variable Value box; 
   otherwise the system does not recognize the folder. For example, type 
   C:\Progra~1\jdk1.6.0 in the Variable Value box.

**Configuring Tomcat**

This procedure assumes that you have downloaded and installed Tomcat on your machine. You can download Tomcat from the Apache website; depending on the version you want to download, you may need to locate the appropriate file in Apache's Archive area. Instructions for downloading and installing Tomcat are also available on the Apache website.

To configure Tomcat, a system variable must point to the folder where you install Tomcat. Installing Tomcat to a simple folder path such as C:\Tomcat makes it easier to define the system variable.

The third-party products discussed below are manufactured by vendors independent of MicroStrategy, and the steps to configure these products is subject to change. Refer to the appropriate Microsoft documentation for steps to access and modify the environment variables.
1. From the **Start** menu, go to **Computer > System properties > Advanced system settings > Environment Variables > System Variables**.

2. Click **New** to create a system variable.

3. In the **Variable Name** field, type **CATALINA_HOME**.

4. In the **Variable Value** field, specify the path of the folder where you installed Tomcat and click **OK**. For example, if you installed Tomcat directly to the C drive, the destination folder is **C:\Tomcat**.

   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.
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.
2. Copy the WAR file to the Tomcat\webapps folder.
3. From the Start menu, select Run.
4. Type cmd in the Open drop-down list and click OK.
5. Browse to the Tomcat\bin folder, where Tomcat is the folder in which you installed Tomcat. For example, in the command prompt, type
   
   cd C:\Tomcat\bin

6. Click Enter. C:\Tomcat\bin> is displayed at the command prompt.
7. Type the required commands to start and stop Tomcat, which depends on your version of Tomcat. For example, for Tomcat 7, type Tomcat7 start to start Tomcat and type Tomcat7 stop to stop Tomcat. Refer to your third-party Apache documentation for information on the commands to start and stop Tomcat.

   If you installed Tomcat under the Program Files folder, type Progra~1
   when you change folders in the command prompt. Otherwise, the system does not recognize the folder. For example, type C:\Progra~1\Tomcat\bin in the command prompt.

Your MicroStrategy JSP application is deployed automatically, based on the following:

- If you have configured Tomcat to deploy an exploded WAR file, which is often the default behavior, a folder is created within the Tomcat\webapps folder:
- When deploying MicroStrategy Web (JSP), the folder is named `MicroStrategy` by default.

- When deploying MicroStrategy Mobile Server (JSP), the folder is named `MicroStrategyMobile` by default.

- If you have configured Tomcat to deploy an unexploded WAR file, the configuration files are created within the system's default temporary file directory. For Windows systems, the temporary file directory is commonly defined by the `TMP` environment variable:

- When deploying MicroStrategy Web (JSP), a `/microstrategy/web-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Web (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.

- When deploying MicroStrategy Mobile Server (JSP), a `/microstrategy/mobile-Version/` folder is created within the temporary file directory, where `Version` is the version number for the MicroStrategy Mobile Server (JSP) product. Within this folder location, various configuration files can be found within the `WEB-INF` folder and its subfolders.

Configuring administrative access your MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must create the users and assign them the role of `admin` under the Tomcat user configuration file. The steps to configure this access are below.
To configure administrative access to your MicroStrategy JSP applications

1. In the Tomcat\conf folder, open the tomcat-users.xml file in a program that allows you to edit the file, such as Notepad.

2. Add the following tag and save the file:

   `<user name="administrator" password="administrator" roles="admin"/>

   You can specify any value in the user name and password fields.

3. Stop and start Tomcat from the command line.

Accessing the MicroStrategy JSP application administrative page

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

1. Access the servlet by typing the following URL in a Web browser:

   - For Web (JSP):
     http://localhost:8080/MicroStrategy/servlet/mstrWebAdmin

   - For Mobile Server (JSP):
     http://localhost:8080/MicroStrategyMobile/servlet/mstrWebAdmin

   The servlet names at the end of the URL are case-sensitive. Make sure to use the correct case when typing the servlet name. If the application server is enabled with security, a dialog box related to the administrator authentication opens.
If you are using Tomcat integrated with IIS, you do not need to specify
the port number in the URL. However, when using Tomcat as a stand-
alone Web container, you must specify the port number. The default
port for Tomcat is 8080.

2. When prompted for a user name and password, use the same values
you specified in the `tomcat-users.xml` file.

3. If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web
Administrator page appears. Add and connect to an Intelligence
Server.

4. If you are deploying MicroStrategy Mobile Server (JSP), the
MicroStrategy Mobile Server Administrator page appears. Add and
connect to an Intelligence Server. Once connected, click **Mobile
Configuration** to configure your MicroStrategy Mobile applications to
communicate with Mobile Server and Intelligence Server. For steps on
how to define this configuration, see the **MicroStrategy Mobile
Administration Guide**. Creating a configuration completes the steps
required to deploy Mobile Server (JSP).

**Launching the project**

In a Web browser, access MicroStrategy Web (JSP) using this URL:

`http://localhost:8080/MicroStrategy/servlet/mstrWeb`

**Deploying with 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).

- **Deploying MicroStrategy Web and Mobile Server, page 237**: Instructions
  for deploying MicroStrategy Web (JSP), Mobile Server (JSP).

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 administration and deployment tools in Oracle Glassfish Server have the same interface regardless of the operating system on which they run. Therefore, the deployment process is the same for all operating systems, and is described below. There are some minor changes in the Windows environment, which are highlighted where necessary.

Launching the Oracle Glassfish Server Administration Console

This procedure describes the steps to launch the Oracle Glassfish Server.

⚠️ The Oracle Glassfish Server is installed. This installation should also include a default domain, commonly named domain1. If you plan to use a different domain, refer to your third-party Oracle documentation for creating a domain.

Copy the WAR file 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 DomainDirectoryDomainName
```

where:

- `DomainDirectory` is the path you defined when creating the domain. You can remove the `--domaindir` option if the domain uses the default directory.

- `DomainName` is the name of the domain you created in the previous steps.

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

```
asadmin start-domain domain1
```

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

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

where:

- `MachineName` is the IP address or the name of the machine where you installed Oracle Glassfish Server.

- `PortNumber` is the port number you provided when creating the domain. The default port number is 4848.

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

**Deploying your MicroStrategy JSP application**

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

- Save the WAR file to the same machine as the Oracle Glassfish Server, or to a
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 237.

To deploy MicroStrategy JSP applications as a WAR file

1. Access the Administration Console by typing the following URL:

   http://MachineName:PortNumber

   where:

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

2. If prompted, type the user name and password that you used to create the domain.

3. Expand the Tree pane on the left side of the Administration Console.

4. Click Applications.

5. Click Deploy.

6. Select Local Packaged File or Directory That Is Accessible from GlassFish Server, and then click Browse Files.
Selecting the WAR file in this manner is recommended as the Packaged File to Be Uploaded to the Server option uploads the WAR file via HTTP, which can require considerable time and system resources.

7. Browse to the location where you saved the MicroStrategy JSP application WAR file.

8. Once you select the appropriate WAR file, click **Choose File**.

9. From the **Type** drop-down list, select **Web Application**.

10. In the **Context Root** field, type the context root for the application, which is included in various URLs for the application:

- The URL to access MicroStrategy Web (JSP)

  
  (http://IPAddress:PortNumber/ContextRoot/servlet/mstrWeb) includes the applications context root, which should be replaced by any name of your choice. For example, you can use the default name of the WAR file, which is MicroStrategy.

- The URL to access the MicroStrategy Mobile Server Administrator Page

  
  (http://IPAddress:PortNumber/ContextRoot/servlet/mstrWebAdmin) includes the applications context root, which should be replaced by any name of your choice. For example, you can use the default name of the WAR file, which is MicroStrategyMobile.

11. In the **Application Name** field, type a descriptive name to distinguish the application from within the Administration Console.

12. In the **Virtual Servers** list, select the appropriate server.
13. Select or clear the additional deployment option check boxes according to your requirements.

   It is recommended you select the Precompile JSPs check box to quickly load the Web pages in the application server when you access it for the first time.

14. Click OK.

Configuring administrative access to MicroStrategy JSP applications

For security purposes, you must only assign certain users the administrative authorization to access the MicroStrategy Web Administrator, Mobile Server Administrator. To do this, users need to be assigned to the mstradmin group, which is part of the admin security role.

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

- File realm
- Administration realm
- Certificate realm

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

   http://docs.oracle.com/cd/E18930_01/html/821-2435/ggkuk.html#gkbiy

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

1. In the Administration Console, from the Tree pane on the left, click **server (Admin Server)**. Ensure that the server is running or click **Start** to start the server.

2. From the Tree pane on the left, go to **Configuration > server-config > Security > Realms** and select file..

3. Click **Manage Users**.

4. Click **New**.

5. Type the following information for the new user:
   - **User ID**: The ID that the user provides when authenticating with the system.
   - **Group List**: The groups that the user is a member of. Type `mstradmin` to provide the user administrative access to MicroStrategy Web Administrator and MicroStrategy Mobile Server Administrator.
   - **New Password**: The password used to authenticate a user.
   - **Confirm New Password**: A confirmation of the password, required when creating a new user.

6. Click **OK**.

7. In the Administration Console, from the Tree pane on the left, select **server (Admin Server)**.

8. Click **Restart**.
Managing the admin security role for specialized group authentication requirements

MicroStrategy provides a descriptor file, glassfish-web.xml, which enables Oracle Glassfish Server to map the existing users or groups to security roles. This file is located within the MicroStrategy JSP application WAR files, and after deployment can be found in the WebApplicationRootDir/WEB-INF folder.

By default, the admin security role and its mstradmin group defined in this glassfish-web.xml file can be used to grant administrative access to the MicroStrategy Web Administrator and Mobile Server Administrator. This provides administrative access without having to make any modifications to glassfish-web.xml. In these scenarios, you can use the steps provided in Deploying your MicroStrategy JSP application, page 238 and Configuring administrative access to MicroStrategy JSP applications, page 241 to complete the deployment and authentication requirements.

While this default behavior supports most deployment requirements, you can modify this glassfish-web.xml file if you have specialized group authentication requirements to use a group other than the default mstradmin group defined for the admin security role. Any groups that are used must be included as part of the admin security role.

Any changes made to the glassfish-web.xml file must be done prior to deploying the MicroStrategy JSP application.

The contents of this file are as follows, which may differ depending on your installation of Oracle Glassfish Server:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Servlet 3.0//EN"
"http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">
<glassfish-web-app>
```
Once you make any changes to this file, you must deploy the application (see *Deploying your MicroStrategy JSP application, page 238*) and assign the security role to the necessary user accounts (see *Configuring administrative access to MicroStrategy JSP applications, page 241*).

Accessing the MicroStrategy JSP application administrative page

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

To access the MicroStrategy JSP application administrative page

1. Access the servlet by typing the following URL in a Web browser:
   
   - For Web (JSP):
     
     \[
     \text{http://IPAddress:PortNumber/ContextRootWeb/servlet/mstrWebAdmin}
     \]

     In the URL listed above, \textit{ContextRootWeb} is the name you provided for the ContextRoot for Web Module box in the section *Deploying your MicroStrategy JSP application, page 238*. 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):
     
     \[
     \text{http://IPAddress:PortNumber/ContextRootMobile/servlet/mstrWebAdmin}
     \]
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 238*. 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 241*.

3. After you are authenticated:

   - If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.

   - If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click **Mobile Configuration** to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the *MicroStrategy Mobile Administration Guide*. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

4. If you are deploying MicroStrategy Web (JSP), proceed to launch the MicroStrategy project. For more information, see *Connecting to the Web (JSP) project page, page 245* immediately below.

**Connecting to the Web (JSP) project page**

After restarting the application server, follow the steps described here to connect to the project page.
To connect to the Web (JSP) project page

In a Web browser, type the following URL:

http://MachineName:PortNumber/ContextRoot

If you have used all the default values, you can access the following URLs:

http://localhost:8080/MicroStrategy/

or

http://localhost:8080/MicroStrategy/servlet/mstrWeb

Undeploying MicroStrategy JSP Applications

Oracle recommends undeploying an application before deploying a newer version. The steps below show you how to undeploy an existing MicroStrategy JSP application, using the Oracle Glassfish Server Administration Console.

To undeploy MicroStrategy JSP applications

1. In the Administration Console, from the Tree pane on the left, click Applications. The Applications page is displayed.

2. Select the check box for the MicroStrategy JSP application.

3. Click Undeploy.

4. After the undeployment is finished, stop and restart the application server for the changes to take effect.

Deploying with 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 12.2.1.3. 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 247**: Default folder structure for each version of WebLogic

- **Preconfiguration Information, page 248**: Configuration that must occur before you begin deploying MicroStrategy Web (JSP) and Mobile Server (JSP).

- **Deploying MicroStrategy Web and Mobile Server, page 249**: Instructions for deploying the application

- **Re-deploy the Application, page 259**: Instructions for re-deploying the application

- **Performance-Based Setup Information, page 259**: Optional configuration settings to increase the application's performance.

The additional configuration steps are not required for MicroStrategy Web (JSP) to run, but these settings can increase its performance. Review the performance-based setup information prior to deploying the system to see if these changes are of interest to you.

### WebLogic Paths and Folder Locations

This section presents the default folder structure for each version of WebLogic, and provides the variable used throughout the rest of this chapter to represent the WebLogic `mydomain` folder path.

Each version of WebLogic is installed with a different default path to the WebLogic `mydomain` folder. When deploying MicroStrategy Web (JSP), you must make some changes within the WebLogic folders. Thus, it is important to understand the WebLogic folder structure for the version of WebLogic you are using. The following path reflects the default folder structure for
WebLogic 12.2.1.3: WEBLOGIC_HOME/user_projects/domains/mydomain/

- WEBLOGIC_HOME is the WebLogic Server home path.
- The folder structures are configurable and your organization may have changed the default names or path.

Throughout this chapter, the WebLogic mydomain folder is referred to as WEBLOGIC_MYDOMAIN_FOLDER. This variable refers to the WebLogic mydomain folder in whatever location it resides on your system. The location of this variable is based on the version of WebLogic and whether your organization has changed the version's default name or path.

Preconfiguration Information

This section provides the preconfiguration information necessary to deploy your MicroStrategy JSP applications on your machine.

This section supports the configuration outlined in the following table. While your setup may vary slightly, for example, you may have different versions of these applications, the overall process remains the same.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Oracle Solaris 10.x or 11.x (on SPARC)</td>
</tr>
<tr>
<td>Web server</td>
<td>Apache 2.x</td>
</tr>
<tr>
<td>Application server</td>
<td>WebLogic 12.2.1.3</td>
</tr>
<tr>
<td>JDK</td>
<td>Oracle JDK 1.6.0 or 1.7.0</td>
</tr>
<tr>
<td></td>
<td>You can download the JDK here.</td>
</tr>
</tbody>
</table>

For information on the version numbers supported or certified by MicroStrategy, see the *MicroStrategy Readme*. 
For information on installing these products, see

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 12.2.1.3, the URL is:

https://docs.oracle.com/middleware/12213/webtier/develop-plugin/overview.htm#PLGWL391

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 259.

Deploying MicroStrategy Web and Mobile Server

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

The Performance-Based Setup Information, page 259 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 250*. 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 253*.

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.

2. [Optional] Rename the WAR file to a name you can easily identify and remember. This name is the context_name used in the uniform resource locator (URL) to access the file.

   If you do not change the name of the file, remember to replace context_name with MicroStrategy when accessing the application from the URL.

To modify the web.xml file for multiple MicroStrategy deployments

If you are deploying more than one MicroStrategy environments on the same WebLogic application server, prior to deployment, you must modify the web.xml file as described below.

1. Unzip the WAR file by using the following command:

   \texttt{jar -xvf FileName.war}

   Where \texttt{FileName} is the name of the WAR file for your MicroStrategy JSP application.

2. Open the \texttt{web.xml} file located in the \texttt{/WEB-INF} directory.

3. Modify the contextPath parameter. By default, this parameter does not have a value. Type a unique string for the value of the contextPath parameter. For example, type WebDep2.

4. Save the \texttt{web.xml} file.

5. Zip the WAR file by using the following command:

   \texttt{jar -cvf FileName.war *}

   Where \texttt{FileName} is the name of the WAR file for your MicroStrategy JSP application.
To deploy the WAR file

1. Transfer the WAR file to the following directory:

   `/WEBLOGIC_MYDOMAIN_FOLDER/autodeploy`

2. The application is automatically deployed. To add and connect to an Intelligence Server, see Configuring administrative access to MicroStrategy JSP applications, page 256.

   To increase the performance of MicroStrategy Web JSP, you can configure additional settings after deployment. For more information, see Performance-Based Setup Information, page 259.

To automatically deploy MicroStrategy JSP applications from an exploded directory

1. Locate the WAR file for your MicroStrategy JSP application.

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:

```bash
# 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:

```bash
# 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 256.

To increase the performance of MicroStrategy Web (JSP), you can configure additional settings after deployment. For more information, see Performance-Based Setup Information, page 259.

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.
   
   If you are deploying more than one MicroStrategy environment on the same WebLogic application server, prior to deployment, you must modify the web.xml file

2. Unzip the WAR file by using the following command:
   
   ```bash
   #jar -xvf FileName.war
   ```
   
   Where FileName is the name of the WAR file for your MicroStrategy JSP application.

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

4. Modify the contextPath parameter. By default, this parameter does not have a value. Type a unique string for the value of the contextPath parameter. For example, type WebDep2.

5. Save the web.xml file.

6. Zip the WAR file by using the following command:
   
   ```bash
   #jar -cvf FileName.war *
   ```
   
   Where FileName is the name of the WAR file for your MicroStrategy JSP application.

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

8. Open the WebLogic Server Administration Console (WLS Admin Console) by typing the following address:
   
   http://IP address:port/console/
where IP address is the IP address of the machine on which you installed the WebLogic application server and port is the port number for the WebLogic application server.

9. Type a valid user ID and password at the prompt. The user ID and password are the ones you specified when installing the WebLogic Server on your machine.

10. To complete this operation, see Configuring from the WebLogic Server Administration Console, page 256.

To manually deploy MicroStrategy JSP applications from the exploded directory

1. Locate the WAR file for your MicroStrategy JSP application.

2. Create a folder in the /WEBLOGIC_MYDOMAIN_FOLDER/autodeploy directory and transfer the WAR file to this directory.

3. Unzip the WAR file using the following command:

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

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

4. Open the WebLogic Server Administration Console by accessing the following address:

   ```
   http://IP address:Port/console/
   ```

   where IP address is the IP address of the machine on which you installed the WebLogic application server and Port is the port number for the WebLogic application server.

5. Type a valid user ID and password at the prompt. The user ID and password are the ones you specified when installing the WebLogic Server on your machine.
Configuring from the WebLogic Server Administration Console

To configure from the WebLogic Server Administration Console, refer to your WebLogic Server Administration Console documentation on steps to install a web application.

Once you have installed the JSP version of MicroStrategy Web and Mobile Server as a WebLogic Server Administration Console web application, you have completed the steps required to deploy the application.

To launch the administrative page for MicroStrategy Web (JSP), Mobile Server (JSP) (JSP), see Configuring administrative access to MicroStrategy JSP applications, page 256.

To increase the performance of MicroStrategy Web (JSP), you can configure additional settings after deployment. For more information, see Performance-Based Setup Information, page 259.

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.
<table>
<thead>
<tr>
<th>Deployment Method</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic deployment</td>
<td>Access the Administrator page from a web browser using this URL:</td>
</tr>
<tr>
<td></td>
<td>• For Web (JSP): <a href="http://IPaddress:7001/context_folder_Web/servlet/mstrWebAdmin">http://IPaddress:7001/context_folder_Web/servlet/mstrWebAdmin</a></td>
</tr>
<tr>
<td></td>
<td>In the URL listed above, context_folder_Web is the name of the folder where the Web (JSP) application was exploded and IPaddress is the IP address of your machine.</td>
</tr>
<tr>
<td></td>
<td>• For Mobile Server (JSP): <a href="http://IPaddress:7001/context_folder_Mobile/servlet/mstrWebAdmin">http://IPaddress:7001/context_folder_Mobile/servlet/mstrWebAdmin</a></td>
</tr>
<tr>
<td></td>
<td>In the URL listed above, context_folder_Mobile is the name of the folder where the Mobile Server application was exploded and IPaddress is the IP address of your machine.</td>
</tr>
<tr>
<td>Manual deployment</td>
<td>Access the Administrator page from a browser using this address:</td>
</tr>
<tr>
<td></td>
<td>• For Web (JSP): <a href="http://IPaddress:7001/Web_name/servlet/mstrWebAdmin">http://IPaddress:7001/Web_name/servlet/mstrWebAdmin</a></td>
</tr>
<tr>
<td></td>
<td>In the URL listed above, IPaddress is the IP address of your machine. Replace the Web_name variable with the name you specified in the deployed name field when configuring Web (JSP) from WebLogic Server Administration Console, for example, MyWebApp.</td>
</tr>
<tr>
<td></td>
<td>• For Mobile Server (JSP): <a href="http://IPaddress:7001/Mobile_name/servlet/mstrWebAdmin">http://IPaddress:7001/Mobile_name/servlet/mstrWebAdmin</a></td>
</tr>
<tr>
<td></td>
<td>In the URL listed above, IPaddress is the IP address of your machine. Replace the Mobile_name variable with the name you specified in the deployed name field when configuring Mobile Server from WebLogic Server Administration Console, for example, MyMobileApp.</td>
</tr>
</tbody>
</table>

2. Type the same user ID and password that was used to start the WebLogic Server on your machine.

In WebLogic, the deployment of a MicroStrategy JSP application automatically associates the WebLogic administrative user with the MicroStrategy JSP application administrator. The WebLogic administrative user is the user who has permissions to start the WebLogic Server on a given machine.
3. If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence Server.

4. If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click **Mobile Configuration** to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the **MicroStrategy Mobile Administration Guide**. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

### Launching the project

The address to launch MicroStrategy Web (JSP) is different for each deployment method. The table below lists the URL you can use to access MicroStrategy Web (JSP).

<table>
<thead>
<tr>
<th>Deployment Method</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic deployment</td>
<td>Access MicroStrategy Web (JSP) from a web browser using this URL:</td>
</tr>
<tr>
<td></td>
<td><a href="http://IPaddress:7001/context_folder/servlet/mstrWeb">http://IPaddress:7001/context_folder/servlet/mstrWeb</a></td>
</tr>
<tr>
<td></td>
<td>where <em>context_folder</em> is the name of the folder where the application was exploded and <em>IPaddress</em> is the IP address of your machine.</td>
</tr>
<tr>
<td>Manual deployment</td>
<td>Access MicroStrategy Web (JSP) from a browser using the address:</td>
</tr>
<tr>
<td></td>
<td><a href="http://IPaddress/name/servlet/mstrWeb">http://IPaddress/name/servlet/mstrWeb</a></td>
</tr>
<tr>
<td></td>
<td>where <em>IPaddress</em> is the IP address of your machine. Replace the <em>name</em> variable with the name you specified in the deployed name field when configuring from WebLogic Server Administration Console, for example, <em>MyWebApp</em>.</td>
</tr>
</tbody>
</table>
Re-deploy the Application

If you have already deployed MicroStrategy Web (JSP) with WebLogic and you change any parameters in the web.xml file, you must re-deploy the application using the WebLogic Server Administration Console. This allows the changes to take effect in the deployed application. To re-deploy MicroStrategy Web (JSP), refer to your WebLogic Server Administration Console documentation on steps to re-deploy (update) a web application.

Performance-Based Setup Information

The performance of MicroStrategy Web (JSP) can be increased by configuring it on various component levels. These additional setup settings are not required, but if you want to increase the performance of MicroStrategy Web (JSP), some changes must be done before or after the deployment procedure. This section provides the following configurations:

Setting the Java Heap Size

The Java heap size for the WebLogic Server can be increased by modifying the MEM_ARGS variable in the startWebLogic.sh script:

1. Open the startWebLogic.sh script from /WEBLOGIC_MYDOMAIN_FOLDER/bin/startWebLogic.sh.

2. Define the following line in the script:

   MEM_ARGS="-Xms512m -Xmx1024m"

   This line reflects an initial Java heap size of 512 MB. MicroStrategy recommends the initial java heap size be set at a minimum of 512 MB, assuming the machine has enough memory space. This value may need to be modified to reflect the requirements of your specific environment. Refer to your third-party application server documentation for information on how to determine a satisfactory Java heap size for your environment.

3. Stop and start the application server.
Precompiling JSP Files

To avoid the time taken to load the Web pages in the application server when you access it for the first time, you must precompile the Java Server Pages (JSP) files before deploying the application. Do this by setting the application server to load all the pages in the application before deployment. Thus, when you connect for the first time, the pages are already loaded and the performance is better.

1. Open the `weblogic.xml` file located in the `/WEB-INF` directory.

2. In the `jsp-descriptor` section, set the `keepgenerated` and the `precompile` parameters to `TRUE`, as follows:

   ```xml
   <jsp-descriptor>
   : 
   : 
   <jsp-param>
   :<param-name>keepgenerated</param-name>
   :<param-value>TRUE</param-value>
   </jsp-param>
   <jsp-param>
   :<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 steps.

Set the `pageCheckSeconds` Parameter

The `pageCheckSeconds` parameter sets the interval, in seconds, at which the WebLogic Server checks to see if JSP files have changed and need recompiling. Dependencies are also checked and recursively reloaded if changed.
You can set the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pages are checked on every request.</td>
</tr>
<tr>
<td>-1</td>
<td>The page is not checked until the server is restarted. Any classes used by the JSP page that live in the servlet classpath are also reloaded.</td>
</tr>
<tr>
<td>n</td>
<td>Interval (in seconds) in which WebLogic Server checks if JSP files have changed. For example, if this is set to 1, WebLogic checks the pages every second to see if the JSP has changed and needs recompiling.</td>
</tr>
</tbody>
</table>

To Set the `pageCheckSeconds` Parameter

1. Open the `weblogic.xml` file located in the `/WEB-INF` directory.

2. In the `jsp-descriptor` section, set the `pageCheckSeconds` parameter value. For example, the following code sets the value to -1:

   ```xml
   <jsp-descriptor>
   
   <jsp-param>
   <param-name>pageCheckSeconds</param-name>
   <param-value>-1</param-value>
   </jsp-param>
   
   </jsp-descriptor>
   
3. Save the file.

Set the WebLogic Reload Period Parameter

In WebLogic, the Reload Period parameter sets how often WebLogic checks whether a servlet has been modified. If the servlet has been modified, WebLogic reloads it. As the MicroStrategy Web (JSP) servlets do not
change after they have been deployed, MicroStrategy recommends that you disable the reload period by setting it to -1. A value of -1 means never reload, and a value of 0 means always reload.

Use the appropriate procedure below, depending on whether you have MicroStrategy Web (JSP) deployed as a duplicate WAR file.

### To Set the WebLogic Reload Period

1. Open the `weblogic.xml` file located in the `/WEB-INF` directory.
2. In the `container-descriptor` section, set the `servlet-reload-check-secs` parameter value. For example, the following code sets the value to -1:
   ```xml
   <container-descriptor>
     <servlet-reload-check-secs>-1</servlet-reload-check-secs>
   </container-descriptor>
   ```
3. Save the file.

### Configuring Apache Web Server to Serve Static Files

Because Web servers are tuned to effectively serve static files, the perceived performance of MicroStrategy Web (JSP) is significantly enhanced if image, style sheet, and JavaScript files are served via the Apache Web server, and the WebLogic Server handles only the servlet requests. Do this by editing two main parameters, `Alias` and `MatchExpression`, in the Apache configuration file `httpd.conf`.

- The `Alias` parameter is used to create a virtual directory in the Apache Web server. The virtual directory is needed to serve static files such as images, style sheets, and JavaScript.
- The `MatchExpression` parameter is used to configure the Apache plug-in so that the WebLogic Server handles only the servlet requests.
To Configure the Apache Web Server to Serve Static Files

1. To change the Alias parameter, add the following lines in the httpd.conf file:

   ```
   Alias /MicroStrategy/images/"/WEBLOGIC_MYDOMAIN_FOLDER/autodeploy/MicroStrategy/images/"
   <Directory "/WEBLOGIC_MYDOMAIN_FOLDER/autodeploy/MicroStrategy/images">
     Options Indexes MultiViews
     AllowOverride None
     Order allow, deny
     Allow from all
   </Directory>
   ```

   These code excerpts assume the application name is MicroStrategy. See Deploying with WebLogic and Apache (Solaris), page 246 for information on default folder structure.

2. Repeat the previous step for the JavaScript and style sheet folders, replacing the word images in the previous code with the folder name where the JavaScript and style sheet files are located.

3. Change the MatchExpression parameter by typing */servlet/* in the MatchExpression parameter. For example,

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

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

The Web server now serves image (GIF), style sheet (CSS), JavaScript, and all other static files, thus reducing the load on the application server and increasing the application’s performance.
Deploying with WebSphere and IBM HTTP Server (AIX)

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

This section includes the following information:

Deploying MicroStrategy Web and Mobile Server

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

The Performance-Based Setup Information, page 272 section provides information on additional settings to increase application performance. These additional settings are not required, but can increase the performance of MicroStrategy Web (JSP).

Launching the WebSphere Administrative Console

The WebSphere Administrative Console can be accessed only if the WebSphere server is started on the machine.

To start the Websphere Application Server

1. Execute the startServer script as follows:

   cd WAS_HOME/bin
   # ./startServer.sh SERVER_NAME

   Typically, server1 is the default server name in WebSphere.

2. Ensure that the Administrative Server has started successfully. Execute the following commands:
To launch the WebSphere Administrative Console

In a browser, type the URL for the administrative console. The URL is of the following form:

http://IP Address:Port/ibm/console

where IP Address is the IP address of the computer on which you installed the WebSphere application server and Port is the port number for the WebSphere Administrative Console. Refer to your third-party WebSphere documentation to confirm the default port number for the administrative console.

Starting the WebSphere default application server

After you launch the WebSphere Administrative Console, you can deploy MicroStrategy Web (JSP) by starting the default application server.

This is applicable for WebSphere Network Deployment Edition or WebSphere Enterprise Edition. For WebSphere Express or WebSphere Base Editions, there is no distinction between an administrative server and a default server. The StartServer.sh command starts the default application server automatically.

To start the default application server

When the WebSphere Administrative Console opens, a tree view is displayed.

1. Expand the Servers node, or click the link to expand the view.

2. Click the Applications Servers link.
3. Select the box next to the application server to start.

4. Click **Start**.

**Installing the Enterprise Application**

**To install the Enterprise Application**

1. Expand **Applications**, and then **Enterprise Applications** to display a list of installed applications.

2. Click **Install**.

**Preparing for the application installation**

The following steps describe the settings that must be specified for the installation.

_copy the WAR file for the MicroStrategy JSP placement to the WAS_HOME/installableApps directory, where WAS_HOME is the WebSphere application server home path._

**To specify settings for the installation**

1. To begin the installation for IBM WebSphere, go to **Applications > Application Types > WebSphere enterprise applications**.

2. Click **Install**.

3. You must specify the path to the WAR file by selecting either the local file system or remote file system option. For local file systems, you can click browse to navigate to the location of the WAR file. For remote file systems, type in the full path for the location of the WAR file.

4. Click **Next**.
5. Select to perform either a **Fast Path** or **Detailed** installation. Either type of installation can support the deployment of MicroStrategy JSP applications.

6. Select the **Generate Default Binding** check box, and ensure that the **Override existing bindings** check box is cleared.

7. Click **Next**.

8. Select the **Precompile Java Server Pages files** check box.

9. Specify the value for the **Directory to Install Application** as `{APP_INSTALL_ROOT}/DefaultNode`

10. Specify an **Application Name** of your choice.

11. Ensure that the **Override class reloading settings for Web and EJB modules** check box is cleared.

12. Click **Next**.

13. Select the **Web Tier** check box and click **Next**.

14. Type a suitable name for **ContextRoot**, which is case-sensitive. Do not include `.war` in the name for **ContextRoot** as this can cause errors when attempting to start the application.

   The URLs to access MicroStrategy JSP applications contain **ContextRoot**, which should be replaced by the name of your choice. For example, MicroStrategy Web JSP uses the URL `http://machine-name/ContextRoot/servlet/mstrWeb` and you can use the default name of the WAR file, which is `MicroStrategy`.

15. Click **Next**.

16. Review the summary and click **Finish**.

17. To grant access to these resources, map the **admin** role to the users
or groups that will be given the administrator privileges for your MicroStrategy JSP application. To access these options in WebSphere, expand the Security options, and then click Global Security.

Security must be enabled for the WebSphere Server for this feature to work.

Regenerating plugin-cfg.xml

1. Expand Environment, and then click Update global WebServer Plugin 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 269.

To restart the application server

The option to stop and start the application server through the administrative console is available only for the Websphere Network Deployment and Websphere Enterprise Editions. To stop and start the application server in Websphere Express and Websphere Base editions, see below.

1. Expand Servers, and then click the WebSphere Application Servers link.

2. Select the box next to the application server you want to stop, and click Stop.

3. Select the application server you want to start and click Start.
To stop and start the application server in Websphere Express and Websphere Base editions, use the following commands:

- `stopServer.sh server1` to stop the application server
- `startServer.sh server1` to start the application server.

Starting a single JSP application

This process starts only a single JSP application, rather than all the applications running on the application server. To stop and start all applications, see *Restarting the application server*, above.

To start the Web module

1. Expand **Applications**, then expand **Application Types**, and then select **WebSphere enterprise applications**.
2. Select the box next to the application to start and click **Start**.

Configuring administrative access to MicroStrategy JSP applications

The administrative pages for your MicroStrategy JSP applications are accessible only to users with an `admin` role. To create the set of users and passwords that are authorized for this access, you must create the necessary role mapping between these users and the `admin` role for the MicroStrategy JSP application. The steps to perform this setup are given above in the section *Preparing for the application installation*, page 266. For more information, refer to your IBM documentation.
To configure administrative access to MicroStrategy JSP applications

1. Access the servlet by typing the following URL in a web browser:
   - For Web (JSP):
     http://IPAddress/ContextRootWeb/servlet/mstrWebAdmin
     In the URL listed above, ContextRootWeb is the name you provided for the ContextRoot for Web Module box in the section Preparing for the application installation, page 266. 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 266. For example, the default name of the WAR file, which is MicroStrategyMobile.

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

2. Type the user ID and password assigned with the admin role.

3. After you are authenticated, add and connect to an Intelligence Server.

4. If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click Mobile Configuration to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on
how to define this configuration, see the MicroStrategy Mobile Administration Guide. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

5. If you are deploying MicroStrategy Web (JSP), proceed to launch the project.

Launching the project

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

To launch the project

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

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

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

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

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

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

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

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

   In these addresses, `ContextRoot` is the name you provided for the context root for Web Module box on Preparing for the application installation page. For example, the default name of the WAR file, which is `MicroStrategy`. For more information, refer to Preparing for the application installation, page 266.

You can now access the MicroStrategy Web (JSP) application.
Uninstalling MicroStrategy JSP applications

You can uninstall the MicroStrategy JSP applications through the WebSphere Administrative console.

1. Go to Applications > Application Types > WebSphere enterprise applications.
2. Select the desired MicroStrategy JSP application.
3. Click Uninstall.
4. Save the configuration in the master repository.

Performance-Based Setup Information

The performance of MicroStrategy Web (JSP) can be increased by configuring it on various component levels. These additional settings are not necessary, but can increase the performance of MicroStrategy Web (JSP).

Setting the Java Heap Size

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

1. Access the Administrative Console.
2. Expand the Servers node.
3. Click the Application Servers link to view the list of application servers.
4. Click the application server name, scroll to Additional Properties and click Process Definition.
5. Click JVM Settings to set the Java heap size settings. MicroStrategy recommends that you initially set the Java heap size to a minimum of 500 MB, assuming the machine has enough memory space.
This value may need to be modified to reflect the requirements of your specific environment. Refer to your third-party application server documentation for information on how to determine a satisfactory Java heap size for your environment.

6. Click **Apply**.

7. Stop and start the application server.

**Precompiling JSP Files**

To avoid the time taken to load the Web pages in the application server when you access it for the first time, you must precompile the Java Server Pages (JSP) files. Precompilation can be done during deployment by selecting the **Enable pre-compile of JSPs** setting. Otherwise, it can be done after deploying the application.

To precompile the JSPs after deployment, set the application server to load all the pages in the application. Then when you connect for the first time, the pages are already loaded and performance is improved.

Before you precompile the JSP files, make sure that:

- The MicroStrategy Web (JSP) application is deployed in the WebSphere environment.

- You know the defined application name and the Web Module's name. You can retrieve these names from the Administrative Console. Locate the application name under the Enterprise Applications node. Locate the Web Module name by expanding the application and clicking **Web Modules**. The default name is **Web Tier**.

---

**To precompile the JSP files**

1. Change the directory to `WAS_ROOT/bin`.

2. Run the following command:
   ```bash
dir/./JspBatchCompiler.sh -enterpriseapp.name
```
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 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

This section provides the preconfiguration information necessary to deploy MicroStrategy JSP applications with Tomcat on your Linux machine.

Installing the JDK

If you have not installed Oracle JDK yet, download the shell file here. Be sure to install the JDK and not the JRE software options.

To configure the JDK, a system variable must point to the folder where you install the JDK. If you install the JDK to a simple folder path such as C:\, setting the system variable is easier and more likely to be correct.

From the location in which to install the JDK, run the file you downloaded:

jdk-Version-linux-i586.bin

For example, to install version 1.6.0, type the following:

jdk-1_6_0-linux-i586.bin

Configuring the JDK

1. Open the /etc/profile file using a program that allows you to edit the file.

2. Add the following line:

   export JAVA_HOME=/PathName/jdkVersion;
where PathName is the destination folder where you installed the JDK and Version is the version, such as 1_6_0, of the JDK.

Installing Tomcat

This procedure assumes that you have downloaded and installed Tomcat in a directory named Tomcat on your machine. If you have not installed Tomcat yet, download the zip file from the following links:

- Tomcat 7.0.x
- Tomcat 8.0.x

Contact your System Administrator or visit the Apache website for instructions on downloading and installing Tomcat.

Configuring Tomcat

After you install Tomcat, you must configure Tomcat. The Tomcat configuration includes creating the environment variable CATALINA_HOME and defining this environment variable to point to the Tomcat directory.

1. Open the etc/profile file in a program that allows you to edit the file.

2. Type the following:

   ```
   export CATALINA_HOME = /PathName
   ```

   where PathName is the directory where you have installed Tomcat.

   For example,

   ```
   export CATALINA_HOME = /Tomcat
   ```

Setting the Java Heap Size

The Java heap size for the Tomcat can be modified by defining the JAVA_OPTS parameter in the catalina.sh file. For example, you can define this parameter as follows:
JAVA_OPTS = "-Xms1024m -Xmx2048m"

This value may need to be modified to reflect the requirements of your specific environment. Refer to your third-party application server documentation for information on how to determine a satisfactory Java heap size for your environment.

Deploying MicroStrategy Web and Mobile Server

After you have performed the configurations described above, you can begin deploying MicroStrategy JSP applications with Tomcat.

Deploying using Tomcat as a standalone Web container

To deploy MicroStrategy JSP applications using Tomcat as a standalone Web container

1. Locate the WAR file for your MicroStrategy JSP application.
2. Copy the WAR file to the Tomcat/webapps directory.

To start and stop Tomcat from the command line

Type `# $CATALINA_HOME/bin/startup.sh` and click **Enter** to start Tomcat, which deploys your MicroStrategy JSP applications automatically, based on the following:

- If you have configured Tomcat to deploy an exploded WAR file, which is often the default behavior, a folder is created within the Tomcat/webapps folder:
  - When deploying MicroStrategy Web (JSP), the folder is named MicroStrategy by default.
  - When deploying MicroStrategy Mobile Server (JSP), the folder is named MicroStrategyMobile by default.
If you have configured Tomcat to deploy an unexploded WAR file, the configuration files are created within the system's default temporary file directory. For Linux systems, the temporary file directory is usually /tmp/ or /var/tmp/:

- When deploying MicroStrategy Web (JSP), a /microstrategy/web-\(\text{Version}\)/ folder is created within the temporary file directory, where \(\text{Version}\) 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-\(\text{Version}\)/ folder is created within the temporary file directory, where \(\text{Version}\) is the version number for the MicroStrategy Mobile Server (JSP) product. Within this folder location, various configuration files can be found within the WEB-INF folder and its subfolders.

Configuring administrative access to MicroStrategy JSP applications

To allow users authorized to access MicroStrategy Web Administrator, MicroStrategy Mobile Server Administrator, you must create the users and assign them the role of admin under the Tomcat user configuration file. The steps to configure this access are below.

1. In the Tomcat/conf directory, open the tomcat-users.xml file using a program that allows you to edit the file.

2. Add the following tags and save the file:

\[
<\text{role rolename="admin"}/>
\]

\[
<\text{user username="admin" password="admin" roles="admin"}/>
\]
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 279.

Accessing the MicroStrategy JSP application administrative page

You can use the steps below to access the administrative page for your MicroStrategy JSP application.

1. Access the servlet by typing the following URL in a Web browser:
   - For Web (JSP):
     http://localhost:8080/MicroStrategy/servlet/mstrWebAdmin
   - For Mobile Server (JSP):
     http://localhost:8080/MicroStrategyMobile/servlet/mstrWebAdmin

   The servlet names at the end of the URL are case-sensitive. Make sure to use the correct case when typing the servlet name. If the application server is enabled with security, a dialog box related to the administrator authentication opens.

2. When prompted for a user name and password, use the same values you specified in the tomcat-users.xml file.

3. After you are authenticated:
   - If you are deploying MicroStrategy Web (JSP), the MicroStrategy Web Administrator page appears. Add and connect to an Intelligence
Server.

- If you are deploying MicroStrategy Mobile Server (JSP), the MicroStrategy Mobile Server Administrator page appears. Add and connect to an Intelligence Server. Once connected, click **Mobile Configuration** to configure your MicroStrategy Mobile applications to communicate with Mobile Server and Intelligence Server. For steps on how to define this configuration, see the **MicroStrategy Mobile Administration Guide**. Creating a configuration completes the steps required to deploy Mobile Server (JSP).

4. If you are deploying MicroStrategy Web (JSP), proceed to launch the MicroStrategy project. In a Web browser, access MicroStrategy Web (JSP) using the following URL:

```
http://localhost:8080/MicroStrategy/servlet/mstrWeb
```

**Deploying MicroStrategy Library**

The sections below provide instructions for setting up different application servers to deploy MicroStrategy Library.

**Deploy MicroStrategy Library on WebSphere**

The steps below apply to WebSphere versions 8.5.5 and 9.

**Disable WebSphere's built-in JAX-RS.**

1. In the IBM WebSphere Application Server, expand **Servers > Server Types > WebSphere Application Servers**.

2. Under Application servers, select `<server name>`.

4. Add a new property.
   
   - **Name:**
     
     com.ibm.websphere.jaxrs.server.DisableIBMJAXRSEngine
   
   - **Value:** true

Add Properties to the Web Container Settings

1. Choose **Servers > Server Types > WebSphere application servers** from the WebSphere main configuration page.

2. Click [YOUR SERVER].

3. Under **Configuration**, choose **Container Settings > Web Container Settings > Web Container**.

4. Choose **Additional Properties > Custom properties**.

5. Click **New**.

6. Add the following fields and click **OK**.
   
   - **Name:** com.ibm.ws.webcontainer.emptyServletMappings
   
   - **Value:** true

7. Click **OK**.

8. Click **Save to the master configuration**.

Add a Specific Web Container Custom Property

1. Choose **Servers > Server Types > Application Servers**.

2. Select the server used for Library.

3. Choose **Web Container Settings > Web Container**.

4. Click **Custom properties**.

5. Click **New**.
6. Enter the property values listed in the following table.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>com.ibm.ws.webcontainer.invokeFlushAfterService</td>
</tr>
<tr>
<td>Value</td>
<td>false</td>
</tr>
<tr>
<td>Description</td>
<td>See the corresponding IBM support page</td>
</tr>
</tbody>
</table>

7. Click OK.

8. Click Save.

9. Restart your application server.

Install and Configure the War File

1. In the Select installation options screen of the WebSphere installation process, make sure to enter the correct path in the Directory to install application textbox and then click Next.

2. On the Map virtual hosts for Web modules screen, select the checkbox next to MicroStrategy Library and click Next.

3. On the Map modules to servers screen, select the checkbox next to your MicroStrategy Library installation and click Next.

4. On the Map context roots for Web modules page, enter /MicroStrategyLibrary as the Context Root then click Next.

5. Select the metadata-complete attribute checkbox next to your MicroStrategy Library installation on the metadata for modules page and click Next.

6. Click Finish and make sure to Save directly to the master configuration.
7. Click the Manage Applications link
   or
   Choose Applications > WebSphere Enterprise Applications.

8. Click Library War.

9. In the Detail Properties section, click Class loading and update detection.

10. Under Class Loader Order, select Classes loaded with local class loader first (parent last).

11. Click OK.

12. In the Modules section, click Manage Modules.

13. Click Library Module.

14. In the Class Loader Order pane, select Classes loaded with local class loader first (parent last).

15. Click OK twice.

16. Click Save.

17. Choose Select check box for MicroStrategyLibrary_war.

18. Click Start.

Review the Server Class Loader Policy

1. Choose Server > Server Types > Application Servers > Your server name.

2. Ensure the Server-specific Application Settings are in one of the following orders:
   - Class loader policy is set to Multiple.
   - Class loader policy is set to Single and the class loading mode is set to parent-last.
- Class loader policy is set to **Multiple** and the class loading mode of all applications, other than Library applications, is set to **parent-first**.

**Deploy MicroStrategy Library on JBoss**

**Before Deploying the MicroStrategyLibrary.war File:**

Open `/jboss-eap-7.0/standalone/configuration/standalone.xml` and search for "jaxrs" and disable following lines:

```xml
<!--
<extension module="org.jboss.as.jaxrs"/>
<!--subsystem xmlns="urn:jboss:domain:jaxrs:1.0"/>
```

**Deploy by Exploded Folder:**

1. **Unzip** MicroStrategyLibrary.war file as a folder, the folder name is MicroStrategyLibrary.war.

2. **Put the** MicroStrategyLibrary.war folder under `.jboss-eap-7.0/standalone/deployments/`.

3. **Download** modules_config.zip and **extract** it under the jboss-eap-7.0/modules folder.

4. **Download** jboss-deployment-structure.xml and **put** it under jboss-eap-7.0/standalone/deployments/MicroStrategyLibrary.war/WEB-INF/.

5. **In the** `.jboss-eap-7.0/standalone/deployments/` folder, **create a marker file named**

   MicroStrategyLibrary.war.dodeploy.
6. After deploying the marker file would be updated to MicroStrategyLibrary.war.deployed, if we want to redeploy the folder, just rename it as before.

7. Open MicroStrategyLibrary/admin page to configure connection to Intelligence Server and Collaboration Server.

Deploy MicroStrategy Library on Wildfly

Before Deploying the MicroStrategyLibrary.war File:

1. Change directories to the folder where you want to unzip the WAR file.

2. Run 'jar xvf xxxxx.war' to unzip the WAR file.

3. Modify the configOverride.properties file with Intelligence Server and Collaboration Server information.

4. Use 'jar -cvf xxxxx.war *' to make the WAR file.

5. Place the new WAR file inside the folder.

Deploy the War File:

1. Stop Wildfly.

2. Put the MicroStrategyLibrary.war file under ..\wildfly-10.1.0.Final\standalone\deployments folder.

3. Restart Wildfly.

Deploy MicroStrategy Library on Jetty

⚠️ Please be aware that Jetty 9 is compatible on Java 8, not compatible on Java 9

Before deploying the .war file security configuration of Jetty is necessary:
1. In the file `[jetty_path]/etc/jetty.xml`, between `<Configure>` markers the file should have `<Call name="addBean">` markers which should read:

```xml
<Call name="addBean"> <Arg> <New
class="org.eclipse.jetty.security.HashLoginService">
<Set name="name">Administrator Authentication Area</Set>
<Set name="config"><SystemProperty
name="jetty.home"
default="."/&gt;/etc/realm.properties</Set> </New> 
</Arg> </Call>
```

2. In the same `/etc` folder add a file `realm.properties` containing the line:

```
admin: admin,server-administrator,content-administrator,admin
```

This will define a user, password, and a set of roles.

To deploy the `MicroStrategyLibrary.war` file:

1. Place the file into `[jetty]/demo-base/webapps`

2. Execute the following command:

```
java -jar ../start.jar --add-to-start=jmx
```

3. Go to the demo-base folder and execute the following command:

```
java -jar ../start.jar
```

Deploy MicroStrategy Library on WebLogic

War file Deployment

1. On a browser, open the WebLogic console.

2. Go to **Deployments > Install**.
3. **Upload** MicroStrategyLibrary.war.

4. Select the **Install this deployment as an application** option.

5. Leave all other settings on the default value and click **Finish**.

6. From the deployed folder, change the necessary properties (Intelligece Server hostname, Intelligence Server port, etc.)

   **Sample deployment folder:**
   
   C:\Oracle\Middleware\Oracle_Home\user_ projects\domains\base_ domain\servers\AdminServer\tmp\appmergegen_ 1500564917141_dossier7391.war\WEB-INF\classes\authConfig.properties

7. Restart the web server.

**Deployment with War file exploded**

1. On the machine where WebLogic is installed, manually unzip the MicroStrategyLibrary.war file to a specific folder.

2. On a browser, open the WebLogic console.

3. Go to **Deployments > Install**.

4. Add the folder where the war is unzipped to the **Path** field.

5. Select the radio button next to `<folder name>` (open directory).

6. Select option **Install as an application**.

7. Select the setting to **I will make the deployment accessible at extracted folder**.

8. Keep other default deployment values and and click **Finish**.

9. After the deployment is successful it should appear as active and
healthy.

Sample deployment result:

http://localhost:7001/MicroStrategyLibrary

To access the URL, you may need to turn off firewall on the corp/labs environment

Using a Response File with Configuration Wizard

As an alternative to stepping through each page of the Configuration Wizard, you can create a response file with the upgrade information and use that response file to automatically upgrade your MicroStrategy systems.

This section provides the following information on using a response file:

Creating a Response File

MicroStrategy recommends that you create a response file through the GUI mode of the Configuration Wizard. You step through the Configuration Wizard and make your selections. Then, at the end of the Configuration Wizard, do not click Finish. Instead, click Save. You are prompted to save your selections in a response file.

You can also create or modify a response file with a text editor. If you do not have access to the GUI mode of the Configuration Wizard, this is the only way to create a response file. MicroStrategy supplies a blank response file template, Response.ini, that you can copy and modify to create your response file. This file is in the Common Files folder of your MicroStrategy installation. By default, this folder is C:\Program Files (x86)\Common Files\MicroStrategy.

For information on all the parameters in the response file, see Response Configuration Parameters and Options, page 292.
Executing a Response File


2. Click Load.

3. Browse to the path where the response file is saved and click Open. The Summary page opens.

4. An overview of all the configuration tasks performed by the response file appears. Click Finish.

To Use a Response File through the Windows Command Line

1. Enter 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.

   A common location of a response file is:

   `C:\Program Files (x86)\Common Files\MicroStrategy\RESPONSE.INI`

   If an error message is displayed, check the path and name you supplied for the response file and make any required changes.

To Use a Response File through the Configuration Wizard in UNIX or Linux

1. From a UNIX or Linux console window, browse to `<HOME_PATH>` where `<HOME_PATH>` is the directory you specified as the Home Directory during installation.

2. Browse to the folder `bin`. 
3. Enter `mstrcfgwiz-editor` and click **Enter**. The Configuration Wizard opens with the Welcome page displayed.

4. Click **Enter**.

5. Enter 1 to select to use a response file, and then click **Enter**.

6. Enter the fully qualified path to the `response.ini` file and click **Enter**.

   /home/username/MicroStrategy/RESPONSE.INI

   If an error message is displayed, check the path and name you supplied for the response file and make any required changes.

---

**To Use a Response File Through the UNIX/Linux Command Line**

1. From a UNIX or Linux console window, browse to `<HOME_PATH>` where `<HOME_PATH>` is the directory you specified as the Home Directory during installation.

2. Browse to the folder `bin`.

3. Enter the following command in the command line and click **Enter**.

   ```bash
   mstrcfgwiz-editor -response /Path/response.ini
   ```

   **Where Path** is the fully qualified path to the response file.

   For example, a common location of a response file is:

   /home/username/MicroStrategy/RESPONSE.INI

   If an error message is displayed, check the path and name you supplied for the response file and make any required changes.
To Use a Response File Through the Windows Command Line

1. Type the following command in the Windows command line:
   ```cmd
   macfgwiz.exe -r "Path\response.ini"
   ```
   Where `Path\` is the fully qualified path to the response file.

   A common location of a response file is:
   ```
   C:\Program Files (x86)\Common Files\MicroStrategy\RESPONSE.INI
   ```
   If an error message is displayed, check the path and name you supplied for the response file and make any required changes.

To Use a Response File Through the Configuration Wizard in UNIX or Linux

1. From a UNIX or Linux console window, browse to `<HOME_PATH>` where `<HOME_PATH>` is the directory you specified as the Home Directory during installation.

2. Browse to the folder `bin`.

3. Enter `mstrcfgwiz-editor`, and click **Enter**. The Configuration Wizard opens with the Welcome page displayed.

4. Click **Enter**.

5. Enter 1 to select to use a response file. Click **Enter**.

6. Enter the fully qualified path to the `response.ini` file and click **Enter**.
   ```
   /home/username/MicroStrategy/RESPONSE.INI
   ```
   If an error message is displayed, check the path and name you supplied for the response file and make any required changes.
To Use a Response File Through the UNIX/Linux Command Line

1. From a UNIX or Linux console window, browse to <HOME_PATH> where <HOME_PATH> is the directory you specified as the Home Directory during installation.

2. Browse to the folder bin.

3. Enter the following command in the command line, and click Enter.

   mstrcfgwiz-editor -response /Path/response.ini

   Where Path is the fully qualified path to the response file.

   For example, a common location of a response file is:

   /home/username/MicroStrategy/RESPONSE.INI

   If an error message is displayed, check the path and name you supplied for the response file and make any required changes.

Response Configuration Parameters and Options

MicroStrategy recommends 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.

   If you are manually editing the response file, make sure to save the file using ANSI encoding.

Upgrading Intelligence Server Components and Projects

The response file parameters in the [UpgradeServer] section upgrade Intelligence Server components and projects. The table below lists the available parameters and the functionality of available options for each parameter.
In this section you can upgrade Intelligence Server components and the projects for an Intelligence Server.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[UpgradeServer]</td>
<td></td>
</tr>
<tr>
<td>UpgradeServer=</td>
<td>Defines whether to upgrade Intelligence Server components and projects, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>* 1: Upgrade Intelligence Server components and projects.</td>
</tr>
<tr>
<td></td>
<td>* 0: Does not upgrade Intelligence Server components and projects.</td>
</tr>
<tr>
<td>UpgradeMD=</td>
<td>Defines whether to upgrade the metadata, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>* 1: Upgrades the metadata repository, which contains the definitions of your MicroStrategy applications and supporting objects. An upgrade of your metadata is required to provide support for all new and updated features in the most recent version of MicroStrategy.</td>
</tr>
<tr>
<td></td>
<td>* 0: Does not upgrade the metadata.</td>
</tr>
<tr>
<td>Upgrade Privileges=</td>
<td>Defines whether to upgrade administrative privileges, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>* 1: Upgrades the privileges used to define the features and capabilities available for each MicroStrategy user. This upgrade is required to reflect any changes in privileges for the most recent version of MicroStrategy. After the upgrade is finished, you can manually modify the privileges for your users and user groups to ensure that all users have access to the required MicroStrategy features. To ensure that the updated user privileges are in compliance with your license agreement, you can use License</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Options</td>
<td>Manager to audit and view user product privileges.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not upgrade administrative privileges.</td>
</tr>
<tr>
<td>MigrateHL=</td>
<td>Defines whether to migrate History List from a flat file to a database, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Migrates your previous file-based History List repository to a database-based History List repository. In pre-9.0 versions of MicroStrategy, all History List files were stored using a file-based method. In addition to the file-based method, it is also possible to store the History List repository in the database. This allows administrators to centrally monitor all History List messages across the system.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not migrate History Lists to a database. Use this option if you did not previously use History List messages or you plan to continue to use a flat file to support and store History Lists.</td>
</tr>
<tr>
<td>UpgradeProjects=</td>
<td>Defines whether to upgrade projects, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Upgrades the projects defined in ProjectLogicalUpgrade.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not upgrade projects.</td>
</tr>
<tr>
<td>ServerUser=</td>
<td>Provide the user name for the MicroStrategy user to connect to the Intelligence Server to upgrade.</td>
</tr>
<tr>
<td>ServerPwd=</td>
<td>Provide the password for the MicroStrategy user to connect to the Intelligence Server to upgrade.</td>
</tr>
<tr>
<td>MDPassword=</td>
<td>Provide the password for the metadata to upgrade.</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PrivilegesSource</td>
<td>If you have previously upgraded the metadata but not upgraded the privileges, provide the version of the metadata that you most recently upgraded your privileges to, from the following list of options:</td>
</tr>
<tr>
<td>Version=</td>
<td>• 9.4</td>
</tr>
<tr>
<td></td>
<td>• 9.5</td>
</tr>
<tr>
<td></td>
<td>• 9.5.1</td>
</tr>
<tr>
<td></td>
<td>• 10</td>
</tr>
<tr>
<td></td>
<td>• 10.1</td>
</tr>
<tr>
<td></td>
<td>• 10.2</td>
</tr>
<tr>
<td></td>
<td>• 10.3</td>
</tr>
<tr>
<td></td>
<td>• 10.4</td>
</tr>
<tr>
<td></td>
<td>• 10.5</td>
</tr>
<tr>
<td></td>
<td>• 10.6</td>
</tr>
<tr>
<td></td>
<td>• 10.7</td>
</tr>
<tr>
<td></td>
<td>• 10.8</td>
</tr>
<tr>
<td></td>
<td>• 10.9</td>
</tr>
<tr>
<td></td>
<td>• 10.10</td>
</tr>
<tr>
<td></td>
<td>• 10.11</td>
</tr>
<tr>
<td>HLTarget=</td>
<td>If performing a History List migration, the type of History List to migrate to:</td>
</tr>
<tr>
<td></td>
<td>• File</td>
</tr>
<tr>
<td></td>
<td>• Database</td>
</tr>
<tr>
<td></td>
<td>• Hybrid</td>
</tr>
<tr>
<td>HLPath=</td>
<td>If performing a History List migration, defines the directory location of the History List flat file to migrate to a database.</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HybridPath=</td>
<td>If performing a History List migration to a hybrid History List, defines the directory location where the hybrid History List files are to be stored.</td>
</tr>
<tr>
<td>HLDSNName=</td>
<td>If performing a History List migration, defines the DSN of the database to migrate the History List tables to.</td>
</tr>
<tr>
<td>HLUserName=</td>
<td>If performing a History List migration, defines the user name to connect to the database for the History List migration.</td>
</tr>
<tr>
<td>HLUserPwd=</td>
<td>If performing a History List migration, defines the password to connect to the database for the History List migration.</td>
</tr>
<tr>
<td>HLPrefix=</td>
<td>If performing a History List migration, defines the table prefix for History List tables.</td>
</tr>
<tr>
<td>HLDropTables=</td>
<td>Defines whether to drop existing tables in the database used for the History List migration, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Deletes existing tables in the database before migrating the History List tables.</td>
</tr>
<tr>
<td></td>
<td>• 0: Appends the migrated History List tables to the database without deleting existing tables.</td>
</tr>
<tr>
<td>HLDeleteFiles=</td>
<td>Defines whether to delete the file based History List after the migration is complete, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Deletes the file-based History List repository once the migration to a database-based History List repository is complete. This option should only be used if the file-based History List repository is no longer needed, and you can ensure the migration process can be completed successfully.</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not delete the file-based History List repository once the migration to a database-based History List repository is complete.</td>
</tr>
<tr>
<td>HLUseUTF8=</td>
<td>Define whether UTF-8 encoding should be used for the History List migration on Windows, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Uses UTF-8 encoding for Windows drivers. This option should be used if your database and History List messages use UTF-8 encoding, and the Intelligence Server is installed on Windows.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not use UTF-8 encoding.</td>
</tr>
<tr>
<td>HLUseUNIXUTF8=</td>
<td>Define whether UTF-8 encoding should be used for the History List migration on UNIX/Linux, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Uses UTF-8 encoding for UNIX/Linux drivers. This option should be used if your database and History List messages use UTF-8 encoding, and the Intelligence Server is installed on UNIX or Linux.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not use UTF-8 encoding.</td>
</tr>
<tr>
<td>HLDB2zOSDBName=</td>
<td>Define the database name to create or migrate History Lists stored in DB2 z/OS. This option should be used if connecting to a DB2 z/OS database.</td>
</tr>
<tr>
<td>HLDB2zOSTableSpaceName=</td>
<td>Define the table name to be used in the database when creating or migrating the History List. This option should be used if connecting to a DB2 z/OS database.</td>
</tr>
<tr>
<td>ThreadNumber</td>
<td>If performing a History List repository upgrade to a hybrid History List, defines the number of database threads to use for the upgrade.</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ProjectLogicalUpgrade=</td>
<td>Defines the projects to update their schedules that send reports and documents to caches, History Lists, and MicroStrategy Mobile as subscriptions. These subscriptions are part of the integrated subscription and delivery features in Intelligence Server 9.0 and later versions. They provide the same ability to send reports and documents to caches, History Lists, and MicroStrategy Mobile as the schedules in MicroStrategy pre-9.0. This option also upgrades the reports and Report Services documents in the projects, which improves performance, reduces storage space requirements, and ensures that graphs display correctly in MicroStrategy Web 10 and Developer 10. To upgrade multiple projects, you must enclose the name of each project in curly braces {} and separate the names with a backslash \ . For example: {Project1}{Project2}...{ProjectN}</td>
</tr>
<tr>
<td>ScheduleUpdate=</td>
<td>Defines the projects to upgrade their schedules to the most recent version of MicroStrategy.</td>
</tr>
<tr>
<td></td>
<td>To upgrade multiple projects' schedules, you must enclose the name of each project in curly braces {} and separate the names with a backslash \ . For example: {Project1}{Project2}...{ProjectN}</td>
</tr>
<tr>
<td>MDXUpdate=</td>
<td>Defines the projects to update their MDX source objects (MDX cubes) that were created in earlier versions of MicroStrategy to take advantage of increased performance and other updates.</td>
</tr>
<tr>
<td></td>
<td>You can integrate MDX sources such as SAP</td>
</tr>
</tbody>
</table>
Options | Description
---|---

**BW, Microsoft Analysis Services, and Hyperion Essbase** into your MicroStrategy environment. For information on integrating MDX sources into MicroStrategy, see the MDX Cube Reporting Guide.

To upgrade multiple projects’ MDX source objects, you must enclose the name of each project in curly braces `{ }` and separate the names with a backslash \. For example:

```
{Project1}\{Project2}\...\{ProjectN}
```

**SysObjIDUnification=**

Defines the projects for which to update system objects so that objects can be moved between unrelated projects using MicroStrategy Object Manager. This is required for projects from pre-9.0.1 versions of MicroStrategy.

To upgrade multiple projects’ system objects, you must enclose the name of each project in curly braces `{ }` and separate the names with a backslash \. For example:

```
{Project1}\{Project2}\...\{ProjectN}
```

**ProjectUpgradeContinueExecuting=**

Defines whether to update the rest of the specified projects if one project update generates an error:

- 1: Continues the update process even if an error is found in one project.
- 0: Halts the update process for all projects if an error is found in one project.

**EncryptPassword=**

Defines whether the password is encrypted in the response file, as determined by the following values:

- 1: Encrypts the password in the response file, which ensures that your password is secure. This
Options | Description
--- | ---
is the default behavior.

- 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.

DB2ZoSDBName= | Defines the database name to update metadatas stored in DB2 z/OS. This option should be used if connecting to a DB2 z/OS database.

DB2ZoSTableSpaceName= | Defines the table name to be used in the database when upgrading the metadata. This option should be used if connecting to a DB2 z/OS database.

LeanObjects= | Defines whether to perform Lean Objects migration at the server level:

- 1: Perform Lean Objects migration.
- 0: Do not perform Lean Objects migration.

LeanObjectsProjects= | Defines whether to perform Lean Objects migration at the project level:

- 1: Perform Lean Objects migration.
- 0: Do not perform Lean Objects migration.

Upgrading Statistics Repositories

The response file parameters in the [UpgradeStats] section upgrade statistics and Enterprise Manager repositories. The table below lists the available parameters and the functionality of available options for each parameter.
# Upgrade Guide

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[UpgradeStats]</td>
<td>In this section you can upgrade statistics repositories to the most recent version of MicroStrategy. You can have more than one [UpgradeStats] section. Additional statistics upgrade sections can be included as [UpgradeStats1], [UpgradeStats2], and so on.</td>
</tr>
<tr>
<td>UpgradeStats=</td>
<td>Defines whether to upgrade the statistics and Enterprise Manager repositories, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>- 1: Upgrades the statistics repository</td>
</tr>
<tr>
<td></td>
<td>- 0: Does not upgrade the statistics repository</td>
</tr>
<tr>
<td>EncryptPassword=</td>
<td>Defines whether the password is encrypted in the response file, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>- 1: Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</td>
</tr>
<tr>
<td></td>
<td>- 0: The password is not encrypted in the response file, which enables you to modify the password in the response file later using a text editor. You can then distribute the response file to multiple users with various login and password credentials. However, be aware that this can compromise your database security if you do not remove the password from the response file before distributing it.</td>
</tr>
<tr>
<td>StatsDSN=</td>
<td>If performing a statistics repository upgrade, defines the DSN of the database that stores the statistics repository.</td>
</tr>
<tr>
<td>StatsUser=</td>
<td>If performing a statistics repository upgrade, defines the user name to connect to the database for the statistics repository.</td>
</tr>
<tr>
<td>StatsPwd=</td>
<td>If performing a statistics repository upgrade, defines the password to connect to the database for the statistics repository.</td>
</tr>
<tr>
<td>StatsPrefix=</td>
<td>If performing a statistics repository upgrade, defines the database prefix for the statistics repository.</td>
</tr>
</tbody>
</table>

## Upgrading an Enterprise Manager Project

The response file parameters in the [EMProjectHeader] section upgrade an Enterprise Manager project. The table below lists the available...
parameters and the functionality of available options for each parameter.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[EMProjectHeader]</td>
<td>In this section you can upgrade an Enterprise Manager project to the most recent version of MicroStrategy. You can have more than one [EMProjectHeader] section. Additional Enterprise Manager project upgrade sections can be included as [EMProjectHeader1], [EMProjectHeader2], and so on.</td>
</tr>
<tr>
<td>EMProject=</td>
<td>Defines whether to upgrade the Enterprise Manager project, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Upgrades the Enterprise Manager project.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not upgrade the Enterprise Manager project.</td>
</tr>
<tr>
<td>EMProjectEncryptPwd=</td>
<td>Defines whether the password is encrypted in the response file, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</td>
</tr>
<tr>
<td></td>
<td>• 0: The password is not encrypted in the response file, which enables you to modify the password in the response file later using a text editor. You can then distribute the response file to multiple users with various login and password credentials. However, be aware that this can compromise your database security if you do not remove the password from the response file before distributing it.</td>
</tr>
<tr>
<td>EMProjectDSSUser=</td>
<td>If performing an Enterprise Manager project upgrade, provides the MicroStrategy user name that can access and administer the Enterprise Manager project.</td>
</tr>
<tr>
<td>EMProjectDSSPwd=</td>
<td>If performing an Enterprise Manager project upgrade, provides the password for the MicroStrategy user that can access and administer the Enterprise Manager project.</td>
</tr>
<tr>
<td>EMProjectDSNName=</td>
<td>If performing an Enterprise Manager project upgrade, defines the DSN of the database that stores the statistics tables and</td>
</tr>
</tbody>
</table>
### Upgrading History List Repositories

The response file parameters in the `[UpgradeHistoryList]` section upgrade a database-based History List repository. The table below lists the available parameters and the functionality of available options for each parameter.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[UpgradeHistoryList]</td>
<td>In this section you can upgrade History List repositories to the most recent version of MicroStrategy. You can have more than one [UpgradeHistoryList] section. Additional History List upgrade sections can be included as [UpgradeHistoryList1], [UpgradeHistoryList2], and so on.</td>
</tr>
<tr>
<td>UpgradeHistoryList=</td>
<td>Defines whether to upgrade a History List repository, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Upgrades the History List repository.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not upgrade the History List repository.</td>
</tr>
<tr>
<td>EncryptPassword=</td>
<td>Defines whether the password is encrypted in the response file, as determined by the following values:</td>
</tr>
</tbody>
</table>

**Options**

- **EMProjectDSNUserName=**  If performing an Enterprise Manager project upgrade, defines the user name to connect to the database for the statistics tables and Enterprise Manager repository.
- **EMProjectDSNUserPwd=**   If performing an Enterprise Manager project upgrade, defines the password to connect to the database for the statistics tables and Enterprise Manager repository.
- **EMProjectDSNPrefix=**    If performing an Enterprise Manager project upgrade, defines the database prefix for the statistics tables and Enterprise Manager repository.
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</td>
</tr>
<tr>
<td>0</td>
<td>The password is not encrypted in the response file, which enables you to modify the password in the response file later using a text editor. You can then distribute the response file to multiple users with various login and password credentials. However, be aware that this can compromise your database security if you do not remove the password from the response file before distributing it.</td>
</tr>
<tr>
<td>HLDSN=</td>
<td>If performing a History List repository upgrade, defines the DSN of the database that stores the History List repository.</td>
</tr>
<tr>
<td>HLUser=</td>
<td>If performing a History List repository upgrade, defines the user name to connect to the database for the History List repository.</td>
</tr>
<tr>
<td>HLPwd=</td>
<td>If performing a History List repository upgrade, defines the password to connect to the database for the History List repository.</td>
</tr>
<tr>
<td>HLPrefix=</td>
<td>If performing a History List repository upgrade, defines the database prefix for the History List repository.</td>
</tr>
<tr>
<td>CopyContent=</td>
<td>If performing a History List repository upgrade, upgrades existing History List messages to the specified format, as determined by the following values:</td>
</tr>
<tr>
<td>1</td>
<td>Upgrades existing History List messages.</td>
</tr>
<tr>
<td>0</td>
<td>Does not upgrade existing History List messages.</td>
</tr>
<tr>
<td>CompressData=</td>
<td>If performing a History List repository upgrade, compresses all existing History List messages, as determined by the following values:</td>
</tr>
<tr>
<td>1</td>
<td>Compresses existing History List messages.</td>
</tr>
<tr>
<td>0</td>
<td>Does not compress existing History List messages. Newly created messages are compressed.</td>
</tr>
</tbody>
</table>
Migrating Web Delivery Subscriptions from Narrowcast Server to MicroStrategy Distribution Services

The response file parameters in the [UpgradeServer] section migrate MicroStrategy Web delivery subscriptions from a Narrowcast Server environment to Distribution Services. MicroStrategy Web delivery subscriptions include email, file, and print subscriptions created from MicroStrategy Web. Migrating these subscriptions from Narrowcast Server to Distribution Services allows the subscriptions to be centralized in Intelligence Server rather than a separate Narrowcast Server. The table below lists the available parameters and the functionality of available options for each parameter.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[MigrateNCS]</td>
<td>In this section you can migrate MicroStrategy Web delivery subscriptions from a Narrowcast Server environment to Distribution Services. You can have more than one [MigrateNCS] section. Additional MicroStrategy Web delivery subscription migration sections can be included as [MigrateNCS1], [MigrateNCS2], and so on.</td>
</tr>
<tr>
<td>MigrateNCS=</td>
<td>Defines whether to migrate MicroStrategy Web delivery subscriptions from a Narrowcast Server environment to Distribution Services, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Migrates MicroStrategy Web delivery subscriptions from a Narrowcast Server environment to Distribution Services.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not migrate MicroStrategy Web delivery subscriptions from a Narrowcast Server environment to Distribution Services.</td>
</tr>
<tr>
<td>DSNNameSBR=</td>
<td>Defines the DSN of the database that stores the Narrowcast Subscription Book Repository.</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UserNameSBR=</td>
<td>Defines the user name to connect to the database for the Narrowcast Subscription Book Repository.</td>
</tr>
<tr>
<td>UserPwdSBR=</td>
<td>Defines the password to connect to the database for the Narrowcast Subscription Book Repository.</td>
</tr>
<tr>
<td>SysPrefixSBR=</td>
<td>Defines the prefix used for the tables of the Narrowcast Subscription Book Repository.</td>
</tr>
<tr>
<td>DSNNameOR=</td>
<td>Defines the DSN of the database that stores the Narrowcast Object Repository.</td>
</tr>
<tr>
<td>UserNameOR=</td>
<td>Defines the user name to connect to the database for the Narrowcast Object Repository.</td>
</tr>
<tr>
<td>UserPwdOR=</td>
<td>Defines the password to connect to the database for the Narrowcast Object Repository.</td>
</tr>
<tr>
<td>SysPrefixOR=</td>
<td>Defines the prefix used for the tables of the Narrowcast Object Repository.</td>
</tr>
<tr>
<td>PSName=</td>
<td>Defines the project source that the Web delivery subscriptions are migrated to.</td>
</tr>
<tr>
<td>DSSUser=</td>
<td>Defines the user name to connect to the project source.</td>
</tr>
<tr>
<td>DSSPwd=</td>
<td>Defines the user password to connect to the project source.</td>
</tr>
<tr>
<td>TargetProjects=</td>
<td>Defines the projects, for a project source, to migrate Web subscriptions from Narrowcast Server to Distribution Services. You can select multiple projects, separating projects by the \ character. For example, TargetProjects=Project1\Project2.</td>
</tr>
<tr>
<td>MigrateWebDeliveryUsers=</td>
<td>Defines whether to migrate all users of Web deliveries to Distribution Services, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Migrates all users of Web deliveries to Distribution Services. Addresses of the MicroStrategy users who</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>subscribed to Web deliveries are assigned to the corresponding MicroStrategy users. No new MicroStrategy users are created.</td>
</tr>
<tr>
<td>Assign Privileges=</td>
<td>If you choose to migrate all users of Web deliveries to Distribution Services, this option defines whether to grant the Use Distribution Services privilege for all Web delivery users that are migrated to Distribution Services, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Grants the Use Distribution Services privilege for all Web delivery users that are migrated to Distribution Services. Ensure that granting these privileges is in compliance with your product license for Distribution Services.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not grant the Use Distribution Services privilege for all Web delivery users that are migrated to Distribution Services.</td>
</tr>
<tr>
<td>MigrateNCSUsers=</td>
<td>Defines whether to migrate Narrowcast Server users and their addresses as contacts in Distribution Services, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>• 1: Migrates Narrowcast Server users and their addresses as contacts in Distribution Services. A contact is created for each Narrowcast Server user. The Narrowcast Server user's addresses are copied as addresses of the contact. A contact is a list of subscription recipient addresses. These recipients do not have the permissions to log in to MicroStrategy Web or Developer. Contacts enable these recipients to receive subscriptions.</td>
</tr>
<tr>
<td></td>
<td>• 0: Does not migrate Narrowcast Server users and their addresses as contacts in Distribution Services</td>
</tr>
</tbody>
</table>
### Options Description

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MigrateTransmittersAction=</td>
<td>These parameters specify the conflict resolution rules when migrating Narrowcast objects to Distribution Services. Conflicts occur when identical objects are present in both the Narrowcast Server system you are migrating from and the Intelligence Server metadata you are migrating to. Defining conflict resolution rules determines how these conflicts are resolved during the migration. You can use the following parameters to define how to resolve any conflicts during the migration:</td>
</tr>
</tbody>
</table>
| MigrateDevicesAction= | • MigrateTransmittersAction for Transmitters  
• MigrateDevicesAction for Devices |
| MigrateSchedulesAction= | • MigrateSchedulesAction for Schedules  
• MigrateSubscriptionsAction for Subscriptions  
• MigrateUsersAction for Users |
| MigrateSubscriptionsAction= | For each object type listed above, you can define conflict resolution rules based on matching IDs. For transmitters, devices, and schedules you can also define conflict resolution rules based on matching definitions. These options are described in the table below.  
In the table below, "source" refers to your Narrowcast Server and "destination" refers to the new metadata tables. |

In the table below, "source" refers to your Narrowcast Server and "destination" refers to the new metadata tables.
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MigrateUsersAction=</td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Match by ID or Definition</td>
<td>Match by IDs, keep existing</td>
</tr>
<tr>
<td>Match by IDs</td>
<td>Match by IDs, replace</td>
</tr>
<tr>
<td>Match by Definition</td>
<td>Match by Definition, merge</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>match. These conflict resolution options are available for transmitters, devices, and schedules.</td>
</tr>
<tr>
<td></td>
<td><strong>Match by Definition, create new</strong></td>
</tr>
<tr>
<td></td>
<td>If objects with matching definitions are found, the matching object in the source is migrated to the destination as a new object. A separate object for the matching object in the destination is also kept. This results in two objects with matching definitions in the destination, which you can modify after the migration is complete.</td>
</tr>
<tr>
<td>EncryptPassword=</td>
<td>Defines whether the password is encrypted in the response file, as determined by the following values:</td>
</tr>
<tr>
<td></td>
<td>1: Encrypts the password in the response file, which ensures that your password is secure. This is the default behavior.</td>
</tr>
<tr>
<td></td>
<td>0: The password is not encrypted in the response file, which enables you to modify the password in the response file later using a text editor. You can then distribute the response file to multiple users with...</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
</tbody>
</table>

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.

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 313).

- 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 315).

- Edit the odbc.ini on Linux. For more information, please see Configuring ODBC Parameters with odbc.ini.

    It is strongly recommended you use the MicroStrategy Connectivity Wizard when creating a new DSN for a MicroStrategy-branded ODBC driver. Use the Microsoft ODBC Data Source Administrator only if you intend to use a driver that is not MicroStrategy-branded.
If you create DSNs using the Microsoft ODBC Data Source Administrator, you must create system DSNs. Otherwise, MicroStrategy interfaces will not recognize them.

Creating a DSN for a data source

If a DSN does not already exist in your empty metadata repository or the repository installed with MicroStrategy, you can add or create a new one.

The MicroStrategy Connectivity Wizard is a tool designed specifically to configure connectivity to data sources by creating a DSN that uses a MicroStrategy-branded ODBC driver.

To create a DSN

1. On Windows, log in to the system as an administrator.
2. From the Start menu, go to Programs > MicroStrategy Tools > Connectivity Wizard.

On Linux:

- Using the Connectivity Wizard interface, perform the following steps:
  - In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation.
  - Browse to the folder bin and type ./mstrconnectwiz, and then press ENTER.
- From the command line, perform the following steps:
  - In a Linux console window, browse to HOME_PATH where HOME_PATH is the specified home directory during installation.
  - Browse to the folder bin and type ./mstrconnectwiz -h, and then press ENTER.
This command displays command line operation syntax and examples for different database platforms. Create your command based on the syntax and examples displayed. Once you perform your command, the DSN is created and no further action is required. For detailed steps on how to use the command line version of this tool, see *Creating a DSN for a Data Source* in *Configuring MicroStrategy Using Command Line Tools*.

3. Click **Next**.

4. Select a database driver with which to create a DSN and click **Next**.

   Only a few databases can contain metadata repositories. For details, refer to the *MicroStrategy Readme*. Only DSNs created to connect to these databases can be used to connect to metadata repositories.

5. Enter the information in the appropriate fields for connecting with the selected database driver. The information to enter varies depending on the database platform that you selected. For more information, see *Creating DSNs for specific data sources*.

6. Click **Test**.

7. Enter the **User Name** and **Password** to connect to the database.

8. Click **Connect**. If the test is performed successfully, the connection with the database is established. If the test fails, verify the correct connection information with your database administrator and make any required changes to the information you provided in the previous steps.

9. Click **Close**, and then **Finish**.

   If you already have an existing DSN with the same name as the one you provided, a message box appears. You have the following options:
Select **Yes** to make sure the DSN points to the location you are expecting. This overwrites the existing DSN.

Select **No** to save the DSN with a different name.

Repeat the above steps to create as many DSNs as you require. At a minimum, create one for your metadata and one for your warehouse.

### Managing ODBC and data sources with Microsoft ODBC Data Source Administrator

The Microsoft ODBC Data Source Administrator manages database drivers and data sources on Windows. The Microsoft ODBC Data Source Administrator utility creates a log with which to trace calls to data sources and to view, create, and edit DSNs. The utility is available from Microsoft and is usually included with the purchase of an ODBC driver.

- It is strongly recommended that you use the Connectivity Wizard when creating a new DSN for a MicroStrategy-branded ODBC Driver. Use the Microsoft ODBC Data Source Administrator only if you intend to use a non-MicroStrategy driver.

- If you choose to create DSNs using the Microsoft ODBC Data Source Administrator, they must be system DSNs. Otherwise, MicroStrategy interfaces cannot recognize them.

---

#### To create a DSN using the Microsoft ODBC Data Source Administrator

1. Log in to the machine as an administrator. This gives you the ability to create a system-wide DSN, rather than a user-specific DSN.

2. In most Windows systems you can access the ODBC Data Source Administrator from the Control Panel. Refer to your third-party Microsoft documentation for steps to access the ODBC Data Source Administrator tool.
3. Click the **System DSN** tab.

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

4. Click **Add**.

5. Select the desired driver and click **Finish**.

   - We recommended that you select a MicroStrategy ODBC driver. These drivers, whose names start with MicroStrategy, were installed when you installed the MicroStrategy application on the computer.

6. Enter the information in the appropriate fields to create a data source for the selected database driver.

   - The information to enter varies depending on the database platform that you selected, which is discussed in *Creating DSNs for specific data sources*.

7. Click **OK**.

## 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 321*, *Creating a database connection, page 325*, and *Creating a database login, page 331*.

When a project architect creates a project, the architect assigns a database instance to that project. A project specifies only one warehouse database
instance at a time, but a database instance can be assigned to multiple projects. Since only one data source can be included in the project's relational schema, all reports and documents return information from a single data source.

If you have a license for the MultiSource Option feature, you can connect a project to multiple warehouse database instances. There can be multiple data sources that connect to the Warehouse Catalog for the project. Since these data source can be integrated as part of the project's relational schema, all reports and documents can return information from multiple data sources. For information on accessing multiple data sources in a project, see the Project Design Guide.

Regardless of whether you have a license for the MultiSource Option, you can also extend a project's access to multiple data sources through other MicroStrategy features. Freeform SQL, Query Builder, and supporting access through MicroStrategy to other MDX cube sources such as SAP BW, Oracle Essbase, and Microsoft Analysis Services allows non-project database instances to be included and used in a project along with the warehouse database instances. For information on Freeform SQL and Query Builder, see the Advanced Reporting Guide. For information on MDX cube sources, see the MDX Cube Reporting Guide.

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

**SQL data warehouses database instances**

A SQL data warehouse database instance is any database instance that connects to a database or other data source through SQL queries. More specifically, this covers database instances used for standard MicroStrategy reporting, Freeform SQL, Query Builder, data marts, and any other relational data source. You can also connect to History List and
statistics tables through SQL data warehouse database instances. The SQL data warehouse database instances are available in the Project Configuration Editor, as shown below.

Selecting a database instance check box makes that database instance available in the project for standard MicroStrategy reporting, data marts, Query Builder, and Freeform SQL. If you have a license for the MultiSource Option, selecting a check box for a database instance also makes the database instance available from the Warehouse Catalog to be part of the project's relational schema.

Database instances can be created as part of the Import Data feature. A database instance used for the Import Data feature is displayed with the icon. These database instances are created with security permissions for the user that created them while using the Import Data feature. If you select one of these database instances to be included as an available database instance in the project, it is recommended that you change the security permissions to a MicroStrategy user with administrative privileges. This includes taking ownership of the database instance and defining an appropriate access control list. This ensures that no changes are made to the database instance by other
users, which could cause a loss of connectivity to the data source. For information on the Import Data feature, refer to the MicroStrategy Web online help.

The shading and color of a database instance in the list of relational database instances reflects how the database instance is being used in the project:

- **Bold text:** The project contains objects that are dependent on the database instance. You cannot choose to disable a database instance that has dependent objects for the project.

- **Normal text:** The database instance is not being used in the project.

Clearing the check box of a database instance removes the database instance from the project and deletes any unused Freeform SQL or Query Builder schemas. You can clear a database instance from a project only if there are no dependent objects in the project for the database instance. For more information on removing a database instance and related Freeform SQL and Query Builder schemas from a project, refer to the System Administration Guide.
MDX cube database instances

An MDX cube database instance is any database instance that connects to an MDX cube source, such as SAP BW, Oracle Essbase, or Microsoft Analysis Services. For information on connecting to and reporting on these MDX cube sources, refer to the MDX Cube Reporting Guide. The MDX cube database instances are available in the Project Configuration Editor, as shown below.

A database instance that has an MDX cube schema is represented with bold text. The shading and color of a database instance in the list of relational database instances reflects how the database instance is being used in the project:

- **Bold**: The project contains objects that are dependent on the database instance. You cannot choose to disable a database instance that has dependent objects for the project.

- **Normal**: The database instance is not being used in the project.
If you remove an MDX cube database instance from a project, you can delete any unused MDX cube schema objects. You can remove database instance from a project only if there are no dependent objects in the project for the database instance. For more information on removing a database instance and related MDX cube managed objects from a project, refer to the System Administration Guide.

For additional information on configuring MDX cube database instances, refer to the MDX Cube Reporting Guide.

MDX schema loading and maintenance

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

- You can choose when an MDX cube schema associated with a database instance is loaded for a project. By default, MDX cube schemas are loaded as needed when MDX cube reports are executed. You can also choose to load MDX cube schemas when Intelligence Server starts. For information on defining when MDX cube schemas should be loaded, refer to the MDX Cube Reporting Guide.

- When you integrate MDX cube sources into MicroStrategy, the data is integrated as an MDX cube schema. Once you integrate an MDX cube source into MicroStrategy, you can exchange the database instance used to connect to the MDX cube schema for a different database instance. This allows you to use different database instances with different login and connection information to access an MDX cube schema. For information on exchanging the database instance used to connect to the MDX cube schema, refer to the MDX Cube Reporting Guide.

Creating a database instance

Database instances are created and modified in the Database Instance Manager, which can be found by expanding Administration for a project.
source, then expanding **Configuration Managers**. When you choose to create a new database instance, the Database Instances Editor opens.

You can also create a new database instance using the Database Instance Wizard that is available in the Database Instance Manager shortcut menu.

The Database Instances Editor has the following tabs:

- **General**—specifies the database instance name, connection type (data source platform or applicable data source), and default database connection.

  The database connection type you choose should match your data source and determines whether the database instance is a relational or an MDX cube database instance.

- **Advanced**—specifies the database name for intermediate table storage if a database other than the warehouse is used to store intermediate tables, as well as other options.
The Advanced tab is not available for MDX cube database instances.

- Job Prioritization—specifies the job prioritization scheme for the instance and the number of prioritized connections.

To create a database instance

1. In MicroStrategy Developer, log in to a project source with administrative privileges.

2. Go to Administration > Configuration Managers > Database Instances.

3. From the File menu, go to New > Database Instance.

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.

  - **Use 3.x ODBC Calls**: The support of ODBC 3.x is introduced in MicroStrategy 9.0. This value is chosen by default for Database Instances in MicroStrategy 9.x and higher. You should use this option if your database management system supports ODBC 3.x. ODBC 3.x will always use extended fetch (SQLFetchScroll) to retrieve blocks of data from the database into memory rather than row-by-row retrieval.
7. On the **Job Prioritization** tab, you can configure how jobs are prioritized for the database instance. For information on configuring job prioritization, see the System Administration Guide.

8. On the **General** tab, in the **Database connection (default)** pane, select the default data source connection and click **OK**.

   If the necessary database connection does not exist, you can create one by clicking **New**. For steps to create a database connection, see *Creating a database connection, page 325* 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 333*.

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.
To create a database connection

Before moving forward, ensure that a database instance has been created as described in *Creating a database instance, page 321.*

1. On the **General** tab, in the **Database connection name** box, type a name to identify the database connection.

2. In the **Local system ODBC data sources** pane, select the data source name for the data source.

3. On the **Advanced** tab, you can define various options per your requirements and the requirements of the database you are connecting to, including:

   - **Database driver mode**: Select one of the following database driver modes:
     - **Multi-process**: Each connection to the warehouse database is spawned as a separate process, identified in Windows Task Manager as `M8DBMPE.exe`. If one process fails, such as when a database access thread hangs or is lost, other processes are not affected.
     - **Multi-threaded**: All connections to the warehouse database are maintained inside the Intelligence Server process `MSTRSVR.exe`. All connections, SQL submissions, and data retrievals from the database are handled within this process.

   MicroStrategy recommends setting all database drivers to multi-process mode. The robustness and stability which come with multi-process mode greatly overshadow any increased efficiency that may come with multi-threaded mode. Problems that appear random and sporadic in multi-threaded operation can often be resolved by switching to multi-process mode.
- **Driver execution mode**: Define the driver execution mode depending on the ODBC driver being used:
  - **Asynchronous Connection**: All statements allocated within the connection should be able to run SQL asynchronously.
  - **Asynchronous Statement**: For each statement, the asynchronous mode is explicitly set.
  - **Synchronous**: Only one statement executes at a time. This is the default value.

  Many newer ODBC drivers do not support asynchronous mode because the driver is capable of opening a new thread and executing a new query while simultaneously running an earlier query. The *MicroStrategy Readme* gives recommendations for the driver execution mode options that can be used for different ODBC drivers.

- **Use extended fetch**: Select this check box to enable Intelligence Server to fetch blocks of data from the database into memory, instead of row-by-row. Be aware that this check box is only available if the database instance is defined to use 2.0 ODBC calls, which is included in the steps *Specifying Warehouse Connection Information, page 316*. When 3.0 ODBC calls are enabled, extended fetch is already enabled and this option is grayed out.

- **Use parameterized queries**: Select this check box to enable Intelligence Server to pass data to the database in blocks instead of row-by-row. For information on how parameterized queries can improve performance in MicroStrategy, see the *Project Design Guide*.

- **Maximum cancel attempt time (sec)**: Defines the maximum amount of time the MicroStrategy Query Engine waits for a successful
attempt before it cancels a query. Values of 0 and -1 indicate no limit.

- **Maximum query execution time (sec):** Defines the maximum amount of time a single pass of SQL can execute on the database. Values of 0 and -1 indicate no limit.

- **Maximum connection attempt time (sec):** Defines the maximum amount of time Intelligence Server waits to connect to the database. Values of 0 and -1 indicate no limit.

- **Additional connection string parameters:** Enables you to pass additional ODBC connection parameters to the database as part of the connection string. This is useful if you need to change ODBC defaults. Click **Preview** to see the entire connection string.

- **Table prefix:** Defines a table prefix that specifies the schema containing the tables to access.

- **Character set encoding for Windows drivers:** The options listed below are only relevant when Intelligence Server is running on a Windows machine:
  - **Non UTF-8** (default): Select this option if the ODBC driver returns information in a character encoding other than UTF-8.
  - **UTF-8:** Select this option if the ODBC driver returns information in UTF-8 character encoding. Drivers for Teradata databases may require UTF-8 encoding.

- **Character set encoding for UNIX drivers:** The options listed below are only relevant when Intelligence Server is running on a UNIX machine:
  - **Non UTF-8:** Select this option if the ODBC driver returns information in a character encoding other than UTF-8.
- **UTF-8** (default): Select this option if the ODBC driver returns information in UTF-8 character encoding. Drivers for Teradata databases may require UTF-8 encoding.

- **Connection Caching**: Specify the caching of the database connection using the following options:
  - **Connection idle timeout (sec)**: Defines the amount of time an inactive connection to the database remains cached until it is terminated. You must also set the Connection lifetime, described above, to a value greater than zero for database connections to be used by more than a single job.

  Enforcement of the connection idle timeout can cause a database connection to be removed before it reaches its connection lifetime. You can use this connection idle timeout to ensure that database connections do not remain in Intelligence Server memory in an idle state for an extended amount of time.

  Enforcement of the connection idle timeout can cause a database connection to be removed before it reaches its connection lifetime. You can use this connection idle timeout to ensure that database connections do not remain in Intelligence Server memory in an idle state for an extended amount of time.

  Enforcement of the connection idle timeout can cause a database connection to be removed before it reaches its connection lifetime. You can use this connection idle timeout to ensure that database connections do not remain in Intelligence Server memory in an idle state for an extended amount of time.

  If you type a value of 0, when the job associated with a database connection is completed, the database connection is deleted and not put into a cache. If you type a value of -1, a database
connection can remain idle and considered for new jobs until the database connection lifetime is reached.

4. On the **General** tab, in the **Default database login name** pane, select the default database login and click **OK**.

If the necessary database login does not exist, you can create one by clicking **New**. For steps to create a database connection, see *Creating a database login, page 331* 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 333*.

You create database logins in the Database Connections dialog box by clicking **New** on the General tab. Any database login created within the Database Connections dialog box is available for use across all database connections in the project source.

MicroStrategy reporting and analysis features require a general set of database login permissions that can connect to and modify the data source and metadata, as described below:

- For the metadata, the Select, Insert, and Update permissions are required. Intermediate tables are created in the metadata for recursive search queries, which requires Create and Drop permissions as well. Updating the schema requires the Delete permission.

- For the data warehouse, the Select, Create, Insert, and Drop permissions are required.
When you choose to create a new database login, the Database logins dialog box opens:

![Database Logins dialog box](image)

To create a database login

Before moving forward, ensure that the following is complete:

- A database instance has been created, as described in *Creating a database instance, page 321.*
- A database connection has been created, as described in *Creating a database connection, page 325.*

1. In the **Database Login** field, type the name of the database login.

2. Provide the user ID and password required to access the data source, using one of the following methods:

   - Type the user ID in the **Login ID** field, and type the password for that user ID in the **Password** field.

   - Select the **Use network login ID** check box to connect to the data source using the network user credentials which are also used to run Intelligence Server. If Intelligence Server is running as a service, this is the user that is running the `mstrsvr.exe` process. To determine this user, in MicroStrategy Service Manager, select **MicroStrategy Intelligence Server** and click **Options**. The user is listed on the Service Startup tab, in the Login field. If the Service
Account Name is defined as System Account, the Windows user credentials are used to access the data source.

3. Click **OK**.

Database logins are passed to the data source any time a user executes a report or browses attribute elements. Therefore, all database logins created in MicroStrategy Developer must be also be created as valid logins in the data source.

User connection mapping

User connection mapping is the process of mapping MicroStrategy users to database connections and database logins. For MicroStrategy users to execute reports, they must be mapped to a database connection and database login.

MicroStrategy users link to database connections and logins using:

- The default database connection (and, therefore, default database login)
- Specialized maps to a database connection and/or database login (different than the default connection and login) for either a user or user group

You can map users to connections and logins in the Project Configuration Editor or Command Manager. For information about how connection maps are used, see the **System Administration Guide**.

MicroStrategy reporting and analysis features require a general set of database login permissions to connect to and modify the data warehouse and metadata, as described below:

- For the metadata, the Select, Insert, and Update permissions are required. Intermediate tables are created in the metadata for recursive search queries, which requires Create and Drop permissions as well. Updating the schema requires the Delete permission.
For the data warehouse, the Select, Create, Insert, and Drop permissions are required.

To create a connection map

Before moving forward, ensure that the following is complete:

- A database instance has been created, as described in *Creating a database instance, page 321.*
- A database connection has been created, as described in *Creating a database connection, page 325.*
- A database login has been created, as described in *Creating a database login, page 331.*

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**.

Minimizing the Impact of Data Changes for Platform Release Upgrades

With each feature release, improvements and enhancements made to the Data Engine can cause minor changes in the data returned when executing dossiers and documents. The impacts of these changes for customers upgrading from one feature release to the next, MicroStrategy 10.10 to MicroStrategy 10.11 for example, are minimal and can be resolved through typical post-upgrade testing.

For customers migrating from one platform release to the next, MicroStrategy 10.4.x to MicroStrategy 2020 for example, the cumulative effect of the changes can have a large impact on your execution results. To minimize these impacts when performing a platform release upgrade, MicroStrategy Workstation allows you to choose the Data Engine version for each project or application in your environment.

Customers Upgrading from MicroStrategy 10.11 or earlier will notice the Data Engine version is **10.4** for all projects in your environment following an upgrade to MicroStrategy 2020. Once the upgrade process is completed, the Data Engine version can be changed to 2020 on a project-by-project basis so that data changes can be identified, evaluated, and resolved while maintaining consistent performance across the rest of the environment. All new projects default to the **2020** Data Engine version.

What does the Data Engine do?

The Data Engine processes queries, against a schema, for a specific data
source, to perform data manipulations.

Processes queries
Attributes, metrics, filters, VLDB settings, security filters, data locales

Against a schema
Attributes, facts, tables, relationships, hierarchies

For a specific data source
SQL, MDX, In-Memory, or a blend of linked datasets

To perform data manipulations
Sorting, slicing, subtotals, smart metrics, derived elements, thresholds, cross-tabbing

Selecting a Data Engine Version

Select a Data Engine version, MicroStrategy Workstation must be connected to
an environment and you must have Administrator privileges.

1. Open the Workstation window with the Navigation Pane.
2. Click Applications to display the projects available in the
   environment.
3. Right click an application and select Properties.
4. Select a version from the Data Engine Version drop-down menu.
5. Click **OK**.

6. Reload the project with MicroStrategy Developer to apply the changes.