TIBCO DataSynapse GridServer® Manager

Installation

Version 7.1.0
July 2022

Document Updated: September 2022

Contents

Typographical Conventions

The following table lists the typographical conventions used in this guide:

Convention	Use
`TIBCO_HOME`	Many TIBCO products must be installed within the same home directory. This directory is referenced in the documentation as *`TIBCO_HOME`*. The default value of *`TIBCO_HOME`* depends on the operating system. For example, on Windows systems, the default value is `C:\tibco`.
`DS_INSTALL`	TIBCO GridServer® installs into a directory within *`TIBCO_HOME`* named datasynapse. This directory is referenced in the documentation as `DS_INSTALL`. The default value of *`DS_INSTALL`* depends on the operating system. For example, on Windows systems, the default installation directory is `C:\tibco\datasynapse`.
`DS_MANAGER`	The *Manager directory* contains the read-only software files; by default, it is a directory within *`DS_INSTALL`* named `manager`, and is referred to as *`DS_MANAGER`*. For example, on Windows systems, the default Manager directory is `C:\tibco\datasynapse\manager`.
`DS_DATA`	The *data directory* is the location of all volatile files used by the application Server such as server properties and configuration. By default, it is a directory within *`DS_INSTALL`* named `manager-data`, and is referred to as *`DS_DATA`*. For example, on Windows systems, the default data directory is `C:\tibco\datasynapse\manager-data`.

Installation Overview

This is your starting point for planning, installing, and configuring TIBCO GridServer® for your site. This guide describes how to install, test, and uninstall the Manager, Drivers, and Engines on Windows and UNIX platforms.

This guide is for a DataSynapse GridServer Manager or technology manager planning or assessing requirements for a GridServer installation and an engineer or administrator installing and configuring a GridServer Manager system.

Preparing for Installation

This section lists prerequisites for a GridServer installation and outlines the installation steps. This section assists developers, system architects, and network architects in planning a GridServer production deployment. Read this section carefully to prepare for installation.

Before you begin your GridServer installation, determine what GridServer components you must install and how to configure your overall system. You must also prepare for the installation, meet system requirements, and have an overall understanding of GridServer and the installation process.

Know the Basics

This guide assumes that you know GridServer concepts. If you do not, see the *TIBCO GridServer® Introducing TIBCO GridServer®* for information about the GridServer component architecture and principles of operation.

You must also be familiar with Windows and UNIX operating systems and TCP/IP networking.

GridServer System Requirements

Ensure that your environment meets the minimum hardware and software system requirements for memory, disk space, and processor speed. See the readme file included in the product archive for the complete list of all hardware and software requirements, including supported operating systems, Java or .NET software, web browsers, and accompanying components. Note that depending on the applications you virtualize, you might have to exceed these requirements accordingly.

Optimize GridServer Manager Architecture

You can deploy the GridServer Manager architecture to give varying degrees of redundancy and load sharing, depending on the available computing resources. To plan an architecture that best supports your needs:

- Determine how your facility wants to use GridServer.
- Estimate expected transaction volume and amount of work.
- Survey hardware and networking requirements for the installation.

About GridServer Manager

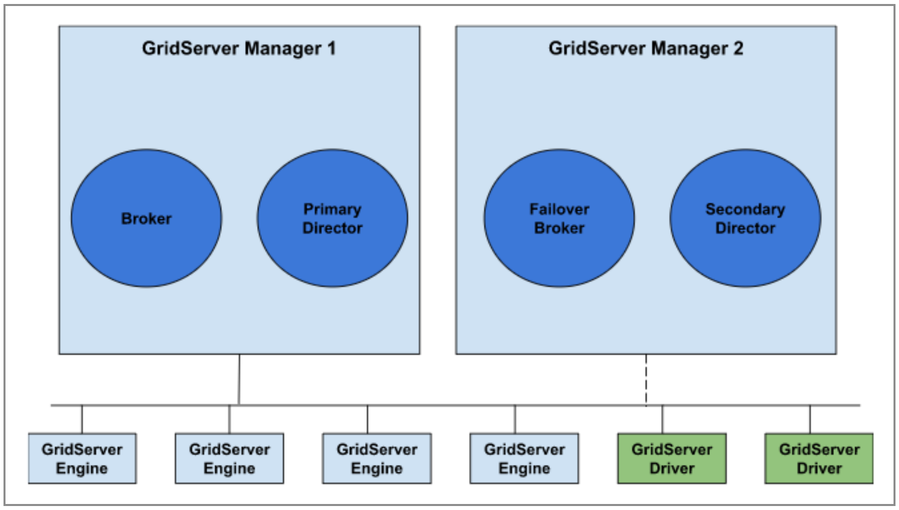
A minimal configuration of GridServer consists of a single Manager configured with a Primary Director and a single Broker. You can add additional Managers containing more Brokers or Directors to address redundancy, volume, or other considerations.

You set an initial configuration of the Manager at the time of installing the Manager, on the **Manager Type** screen. To change the configuration of a running Manager, go to **Admin > System Admin > Manager Reinstall**, follow the screens for Manager reinstallation, and restart the Manager when the configuration is complete.

Redundancy

If you have a minimal configuration including a single Director and single Broker, Engines and Drivers log in to the Director. In this configuration, if the Director fails (as in the case of hardware or network failure), Drivers or Engines cannot establish new connections.

To prevent this problem, run a second Manager with a Secondary Director, and configure Engines and Drivers with the address of both Directors. If the Primary Director fails, Engines and Drivers contact the Secondary Director, which routes Engines and Drivers to Brokers in the same manner as the Primary Director. The following figure shows an implementation with two Managers.

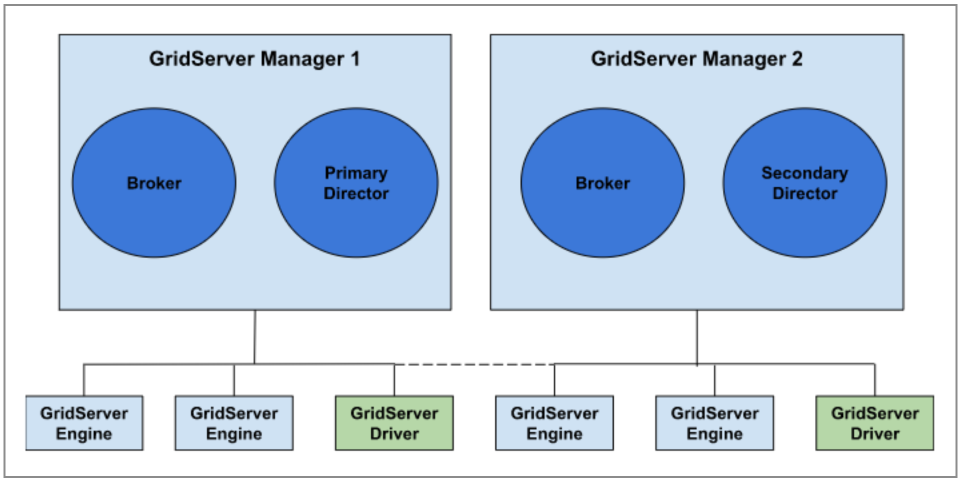


In this typical GridServer installation, there is a Primary Director and a Secondary Director installed on two separate systems. On the same system as the Primary Director, there is a Live Broker. A Failover Broker is installed on the same system as the Secondary Director. Add Engines and Drivers to the cluster as desired. This configuration is fully redundant but does not do any load sharing.

In addition to redundant Directors, a Broker can have a backup on another Manager. You can designate a Broker to be a Failover Broker on a Manager during installation or reinstallation. If no other regular Brokers are available (because of a failure), Directors temporarily route Drivers and Engines to Failover Brokers. The figure to the left shows a Failover Broker on the second Manager.

For more information about redundancy, see the *TIBCO GridServer® Administration*.

Grid Size



This GridServer configuration shows a Primary Director and a Secondary Director on separate systems and live Brokers on both systems. You can add Engines and Drivers to the cluster as desired. This configuration is redundant.

In larger grids, the number of Engines in the grid can require more capability than a single Broker can provide. To distribute load, add more Brokers to other Managers at installation. For example, the figure to the right shows a two-Manager system with two Brokers. Directors route Engines and Drivers to Brokers in round-robin fashion.

Component OS and Version Interoperability

Running Brokers on the same OS as other components is not a requirement and gives no performance benefit. However, running the same OS for all components does simplify administration and troubleshooting.

A GridServer installation typically requires that GridServer components all reflect the same version. When you upgrade a GridServer Manager, it is best to upgrade all Drivers. Engines upgrade themselves automatically.

Other Considerations

Several other factors can influence how you integrate GridServer with your computing environment:

- Instead of using one grid for all types of Services, you might prefer to divide different subsets of Services (for example, by size or priority) to different Directors.
- Your network can dictate how to plan your Manager environment. For example, if you have offices in two parts of the country and a relatively slow extranet but a fast intranet in each location, you can install a Manager in each location.
- Different Managers can support data used for different Service types. For example, you can configure one Manager for Services accessing a SQL database and another Manager for Services that don’t access the database.

With this flexibility, it’s possible to plan a Manager model to provide a work space that facili-tates your workload and traffic. For more information about designing your Manager environment, contact our Integration Services staff, and we can help you determine how to best configure your installation.

Configure Your Network

Since GridServer is a distributed computing application, successful deployment depends on your network configuration. Treat GridServer Managers the same way you treat your other mission-critical file and application servers: assign GridServer Managers static IP addresses and resolvable DNS hostnames. You can configure GridServer Engines and Drivers in several different ways. To receive the full benefit of peer-to-peer communication you must enable communication between Engines and Drivers (the default), but you can also configure GridServer to work with a hub-and-spoke architecture by disabling Direct Data Transfer.

Name Service

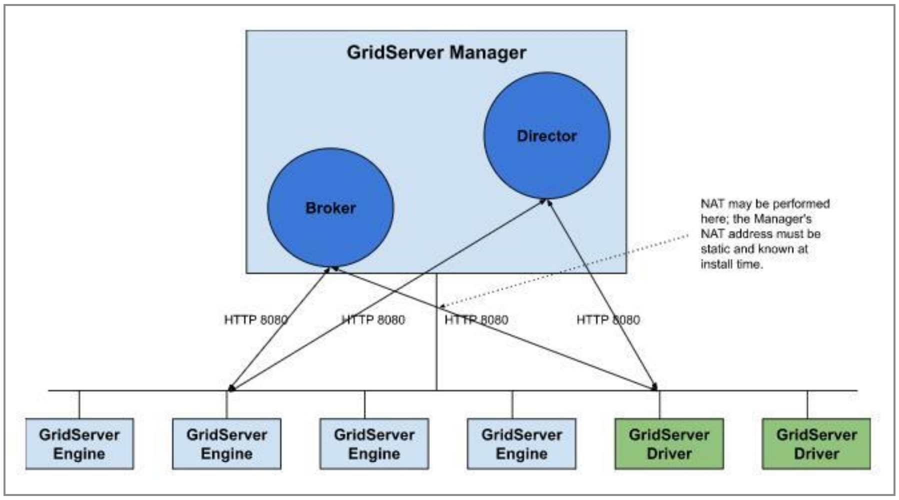
Run GridServer Managers on systems with static IP addresses and resolvable DNS hostnames. In a pure Windows environment, it is possible to run GridServer using just WINS name resolution, but this is not recommended for larger deployments or heterogeneous environments. GridServer does not support DHCP-assigned client addresses for Manager components. DHCP-assigned addresses are acceptable for Engines and Drivers.

Director-Broker Communication

Directors and Brokers communicate using TCP on port 5635. This port is used for both initial login and further communication. If you need to specify a different port (for example, if you are running multiple instances on a single machine), you can do so at installation, or by reinstalling the Manager at **Admin > System Admin > Manager Reinstall**.

Manager-Engine and Driver-Manager Communication

All communication between Engines and Managers (Directors and Brokers) and between Drivers and Managers uses the HTTP protocol. The Engine or Driver acts as HTTP client and the Manager acts as HTTP server.



All communication between Engines and Managers or Drivers and Managers is with the HTTP protocol on the port assigned when the Manager is installed. In this diagram we assume the assigned port is 8080. All HTTP communication is initiated by the Engine or Driver as the HTTP client.

An Engine Daemon periodically checks if its IP address has changed. If so, the Engine Daemon restarts itself and all Engine instances so that they use the new IP address.

Configuring the Manager by Using NAT

You can configure the Manager to work with a NAT device between the Manager and the Engines or Drivers. If the untranslated site of the network has no Engines, you can configure to work with a NAT device by specifying the external (translated) address of the NAT device when referring to the Manager address in the Manager URL field during Manager installation, in Driver installation, and in Windows and Linux Engine installation.

If clients (Engines and Drivers) exist on both sides of a NAT device, you can configure NAT Translation on the Manager. This enables the URLs used for Manager-to-client communication.

Configure NAT Translation in the GridServer Administration Tool

1. Go to **Admin > System Admin > Manager Configuration > Communication**.
2. Find the **NAT Translation** heading.
   The first field, **NAT Translation Range**, specifies the range of IP addresses within the NAT untranslated network to be translated.
3. In the **NAT URL** field, type the external URL to use for the Manager on the translated side of the network. If you want the NAT URL used for clients inside the NAT Translation range, set **Translate Inside Range (True/False)** to **True**; to use NAT translation for clients outside the NAT Range, set this to **Outside**.

Using a VPN

Engines and Clients can run behind a VPN, provided that the Engine Daemon or Driver can determine the correct IP address to use. If VPN client software on the Engine/Client machine handles the VPN, you can possibly specify which IP address to use. For Engines, in the Engine Configuration, use the **Net Mask** setting to force the engine to pick the VPN interface. For Clients, you can set the `DSLocalIPAddress` property to the IP number of the correct interface.

If an external device like a router handles the Engine/Client-side VPN, it is unlikely that the Engine Daemon or Driver can determine the VPN address. Running Engines or Clients in this configuration is not possible.

Resource Synchronization

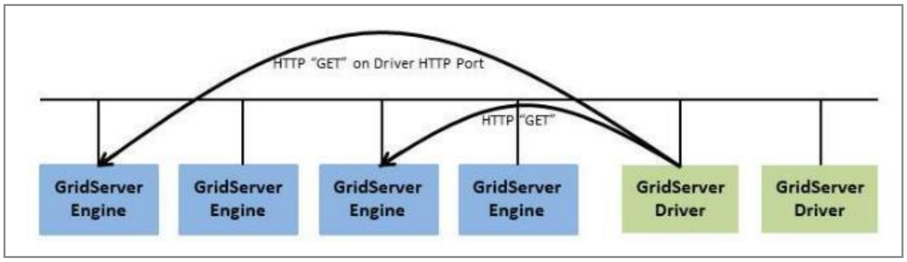
Normally, GridServer synchronizes resources from Directors to Brokers, then from Brokers to Engines. However, in situations when you want only to deploy resources to a subset of Engines, you can disable resource synchronization in the Engine Configuration, and then set the lib directory to point to a shared directory containing resources. On the Engine Configuration page, set the **Synchronize Resources** property in the **Resource Validation** section to false to disable resource synchronization for that Engine Configuration.

Consider the following business use case: using offsite rented systems (on-demand systems) for Engines. An offsite shared collocation facility hosts some Engine systems. You do not fully trust the colocation provider to erase all sensitive data before reusing the systems, so you keep all resources, including code, on a file server at your site. The offsite Engines have a special offsite Engine Configuration pointing lib directories to a file server at your site. You configure other Engines at your site to use resource deployment in the normal way.

Direct Data Transfer

By default, GridServer uses Direct Data Transfer, or peer-to-peer communication, to optimize data throughput between Drivers and Engines. Without Direct Data Transfer, all task input and output goes through the Manager. Sending the input and output through the Manager uses more memory and disk on the Manager and results in lower throughput.

Using Direct Data Transfer, the Driver and Engine nodes do the “heavy lifting,” and only lightweight messages go through the Manager. Direct Data Transfer requires each peer to know the IP address that it presents to other peers. In most cases, therefore, Direct Data Transfer precludes the use of NAT between the peers. Likewise, Direct Data Transfer does not support proxies.

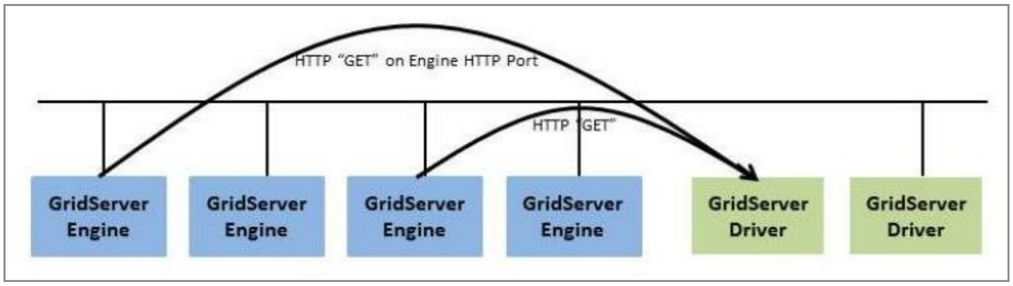


When using Direct Data Transfer to move TaskInput data from Drivers to Engines, the Engines pull the data via HTTP directly from the Driver. You can configure the Driver's embedded HTTP server to use a static TCP port or an ephemeral TCP port according to the `driver.properties` file.

You can also install an optional Engine Hook to enable communication between Drivers and Engines when NAT is in use.

For GridServer deployments that use NAT and do not have the optional Engine Hook, you can support NAT between Drivers and Engines by disabling peer-to-peer communication in one of two ways:

- If, from the perspective of the Drivers, the Engines are behind a NAT device, the Engines cannot provide peer-to-peer communication. In this case, disable Direct Data Transfer in the Engine configuration.
- If, from the perspective of the Engines, the Drivers are behind a NAT device, then the Drivers cannot provide peer-to-peer communication. Provide the Driver addresses if you know them in advance. Otherwise, disable Direct Data Transfer in the Driver properties file.



When using Direct Data Transfer to move TaskOutput data from Engines to Drivers, the Engines pull the data via HTTP directly from the Engines.

SSL

Depending on your application server, you can configure your Manager to use SSL selectively or for all component communication and administration.

For information about using SSL, see Configuring SSL.

Install the GridServer Components

After preparing for your installation, install the GridServer components. This section introduces the components and directs you to their installation instructions.

GridServer Manager

To install a Manager, follow the installation procedure in the Manager Installation section. You must have a running Manager before you install an Engine.

For more information about installing Managers, see Installing DataSynapse GridServer.

GridServer Engine

GridServer Engines come in two variants: Windows Engines and UNIX Engines.

- The Windows Engine is packaged as an MSI installer. It can be manually installed or network installed using a tool such as SMS.
- UNIX Engines are packaged as `tar.gz` archives.

Security Considerations

On the Windows platform, Engines and Managers run as services owned by a user specified during installation.

To prevent unauthorized users from accessing files in the Engine’s directory tree, the Windows system administrator can set up the Engine directory to

- Not inherit permission from the parent directory and
- Grant full access to the built-in `SYSTEM` and `SERVICE` users but not grant access to any other users

On UNIX platforms, Engines and Managers install with the file permissions of the user that installs them. That same user must start the Engines and Managers. For this reason, choose a Manager port above 1024.

For directions on installing Engines, see Windows Engine Installation, UNIX Engine Installation and Enabling Engine Daemon Authentication.

GridServer Driver

Application programs use the GridServer Driver to communicate with the GridServer Manager and thereby leverage the grid’s compute resources. Your application deployment usually bundles the Driver deployment. For more information about installing the GridServer SDK, which includes all Drivers, see the *TIBCO GridServer® Developer's Guide*.

SpeedLink

SpeedLink is a high-throughput extension to GridServer, and is included in the GridServer SDK. For more information, see the *TIBCO GridServer® Speedlink.* .

Database

You can also configure GridServer to use an external reporting database, on an enterprise database system. For more information about database configuration, see Configuring a Reporting Database.

Manager Installation

To begin your DataSynapse GridServer installation, you must install one or more Managers. This section describes how to install the Manager and its components such as Primary and Secondary Directors, and Standalone and Failover Brokers.

This section assumes that you are familiar with the previous section, “Installation Overview” and know which components you want to install on each Manager.

Installing DataSynapse GridServer

When upgrading from a previous version of GridServer, see Upgrading GridServer and complete all steps before beginning the installation process in this section.

Task A: Copy Files Before Installation

Before you install TIBCO GridServer®, you must decide where to install the Manager. TIBCO software is installed in a TIBCO home directory, which is referred to in this documentation as TIBCO_HOME and is typically `c:\TIBCO` for Windows, or `/opt/TIBCO` for UNIX. TIBCO GridServer® is installed in an *installation directory* within this, which is referred to as DS_INSTALL and is typically TIBCO_HOME`/datasynapse`. A Manager installation has files in two locations:

- The *Manager directory* is the location of all static, non-volatile files, and is referred to as DS_MANAGER. The recommended Manager directory location is DS_INSTALL`/manager`. Do not install the Manager in a directory with a name containing a space, such as `c:\Program Files`. Also, do not install the Manager in a directory that is a symbolic link to another directory.
- The *data directory* is the location of all volatile files used by the application server, such as server properties and configuration. This is referred to as DS_DATA. By default, these files are installed in a directory created at `DS_INSTALL/manager-data`. To change this location, see Running Managers With a Different Data Directory.

  On UNIX systems, the `server.sh` script attempts to create the data directory if it does not already exist. If this script fails because it can’t create the data directory, you must either run the script as root, create the data directory with write permissions for the installation user before you run the script, or set the data directory to another location.

> ⚠ **Warning**
>
> Do not set the data directory to be the same as the installation directory.

To copy files used to install TIBCO GridServer®:

1. The software ships as a gzipped TAR archive and a JAR file of third-party libraries. The gzipped TAR archive is unpacked in the installation directory. To unpack it on a Windows system, use WinZip or a similar tool. On a UNIX system, use `tar`. For UNIX, use the following command:

   ```
   tar -xvzf TIB_GridServer*gz
   ```

> ℹ **Note**
>
> The TAR file contains the `datasynapse` directory. If you, for example, would like your installation directory to be `/opt/TIBCO/datasynapse`, expand the archive in `/opt/TIBCO`.

2. If you plan on using Windows Authentication, after unarchiving the installation archive, install the third-party library JAR with the following command, where *version* is the version of GridServer you are installing, and *path* is the DS_MANAGER path:

   ```
   java -jar TIB_gridserver_version-lgpl.jar path
   ```

3. Install the unlimited strength JCE for your Java SDK. The files reside in `DS_MANAGER/webapps/livecluster/WEB-INF/etc/jce`. Follow the instructions in the `README.txt` for your SDK to install the files.

Running Managers With a Different Data Directory

All volatile files used by GridServer are stored in the data directory, which by default is `DS_INSTALL/manager-data`. You can change this location by setting the data directory.

To set your data directory:

Procedure

1. Unpack the installation in the installation directory.
2. Prior to running your Manager for the first time, set the `DS_DATA_DIR` environment variable to the location of the data directory.
3. Run the `server.bat` or `server.sh` script with the `prepare` argument to copy volatile files into the data directory.

When specifying a data directory, it cannot contain any spaces in its name, be the same directory as the installation directory, or be a child directory of the `DS_MANAGER/webapps/livecluster` directory. Also, the installation and data directory cannot be on two different Windows drives (for example, `C:` and `T:`).

Task B: Configure Server Settings

If you need to make changes to any of the following, make the changes to the appropriate config file or environment variable, as required:

Java Settings

You must set the `JAVA_HOME` environment variable to the root directory of your Java installation.

Windows

To set the `JAVA_HOME` environment variable:

1. Right click My Computer and select Properties.
2. On the Advanced tab, select Environment Variables, and then add or edit `JAVA_HOME` to point to where the Java JDK installation is located, for example, `C:\Program Files\Java\jdk1.8.0`.

UNIX

To set the `JAVA_HOME` environment variable, if the Java installation is located in *java-install-dir*:

- Korn and bash shells:

```
export JAVA_HOME=java-install-dir;
export PATH=$JAVA_HOME/bin:$PATH
```

- Bourne shell:

```
JAVA_HOME=java-install-dir;
export JAVA_HOME; PATH=$JAVA_HOME/bin:$PATH; export PATH
```

- C shell:

```
setenv JAVA_HOME java-install-dir;
setenv PATH $JAVA_HOME/bin:$PATH;
export PATH=$JAVA_HOME/bin:$PATH
```

Data Directory

The data directory is the location of volatile files used by the application server, such as server properties and configuration. For more information, see Running Managers With a Different Data Directory.

Port Changes

By default, the Manager uses HTTP on port 8000 for messaging, port 8080 for the web-based Administration Tool, and port 8005 for Tomcat management. You can change any of these locations by editing the `server.xml` file in the DS_DATA`/conf` directory.

For more information, see Changing the HTTP Ports.

HTTPS

By default, all communication takes place over HTTP; you can configure the Manager to use HTTPS instead.

For more information, see Configuring SSL.

Default File Handle Limit

If you are installing a Manager on a UNIX system and plan to run a large grid, you might need to increase the default limit for the number of open file handles. The default on most

UNIX systems is 1024, which is insufficient for a busy grid. In Linux, this is done in the `/etc/security/limits.conf` file.

Task C: Start the Manager

Start the Manager according to instructions in the Windows or UNIX section that follows.

Windows

Start the Windows Manager by running the `server.bat` file, located in the DS_MANAGER directory of the GridServer distribution. To test, open a command window and run `server.bat run` to launch the Manager and start GridServer. The command window contains log messages from the application server.

The `server.bat` script takes the following arguments:

Argument	Description
`start`	Start the Manager. The first time this runs, it copies volatile files into the data directory.
`prepare`	Copy volatile files into the data directory without running the server.

The Manager is typically run as a Windows Service in production systems. To do so, open a command window as administrator and run `service.bat` with the following arguments:

Argument	Description
`install`	Install the Manager as a Windows service.
`remove`	Remove the Manager’s service.
`start`	Start the installed service.
`stop`	Stop the service.
`--user` *username password*	Can be specified to use a Windows service as another user.

Argument	Description
	For example, to install as user `jsmith` with password `test123`: `service.bat install --user jsmith test123`

For initial installation, install the Manager as a Windows service, then start the service.

UNIX

Launch the Manager by invoking the `server.sh` script, located in the DS_MANAGER directory of the GridServer distribution, with the following arguments:

Argument	Description
`run`	Start the Manager in the foreground. The first time this runs, it copies volatile files into the data directory.
`prepare`	Copy volatile files into the data directory without running the server.
`start`	Start the Manager in the background.
`stop`	Stop the Manager running in the background

You can configure the Manager to launch automatically at system startup and stop cleanly at shutdown. Since the `server.sh` script conforms to the standard `start/stop` argument convention (“rc script”), you can accomplish this simply by linking to the appropriate files in the `/etc/rc.d` directories. For more information about initialization and termination scripts, refer to the `init` and `init.d` man pages on your UNIX system and see A Sample UNIX rc.d Script

Note

When installing the Manager on cloud (AWS), you might have to perform a few additional actions to improve the performance of your system.

If you are likely to launch a large number of Engines simultaneously, you must apply AWS auto-scaling concepts. Engine installation steps are the same as those described earlier. You can then perform the following steps:

1. Set `JAVA_HOME` to the latest version of JAVA being used by GridServer.

2. Run the following commands to set machine-level properties such as `maxThreads`:

   ```
   sudo ulimit -c unlimited

   sudo sysctl -w net.core.somaxconn=1024

   sudo sysctl -w net.core.netdev_max_backlog=5000

   sudo sysctl -w net.ipv4.tcp_max_syn_backlog=5000

   sudo sysctl -w fs.file-max=100000
   ```

3. In the GridServer Administration Tool, navigate to **Broker > Broker config** and change the maximum number of Engines as per your requirement.

4. Go to *DS_DATA*`/conf/` and update the `server.xml` file. Change `maxThreads` and `acceptCount` as per your requirement.

In the following example, we have set the value of `acceptCount` and `maxConnections` as 500:

```
<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" acceptCount="500" maxConnections="500"/>
```

After the above changes, you must restart the Manager.

The above steps might vary for other cloud service providers such as Azure or GCP.

Task D: Initialize Your Manager

After the Manager is running for the first time, you must initialize it.

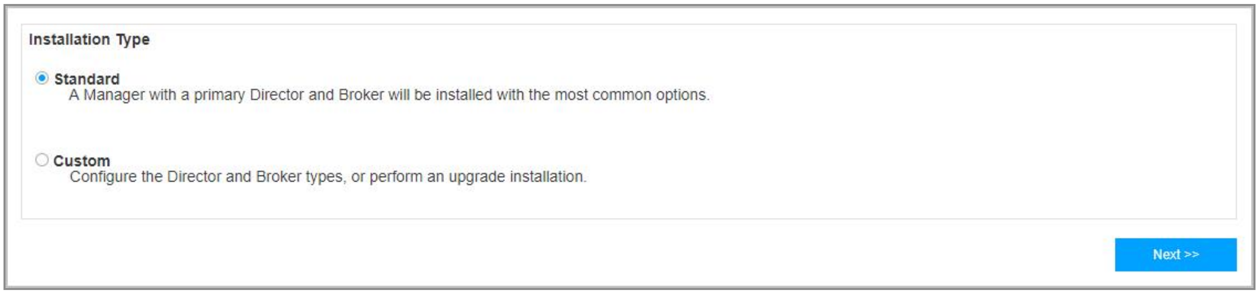
To initialize the Manager:

1. Start the GridServer Administration Tool.

   Go to `http://`*yourhost*`:`*port* to open the installation page.

2. Read and accept the End User License Agreement (EULA) to continue installation.

3. Select Installation Type.



Installation Type

● Standard
A Manager with a primary Director and Broker will be installed with the most common options.

○ Custom
Configure the Director and Broker types, or perform an upgrade installation.

Selecting Installation Type

   Choose a **Standard** or **Custom** Manager installation. The default, **Standard**, installs a primary Director and a Broker with all default settings. If you select **Standard**, click **Next** and skip to step 4 in this procedure.

   Choose **Custom** to install a new Manager (for instance to install a secondary Director, failover Broker, or no Broker) or upgrade an old installation.

4. Enter Configuration Values.

   In the GridServer Manager Configuration page, select **New Manager Installation**, click **Next**, and enter values for each applicable item on each screen. For information about any page, click **Page Help**.

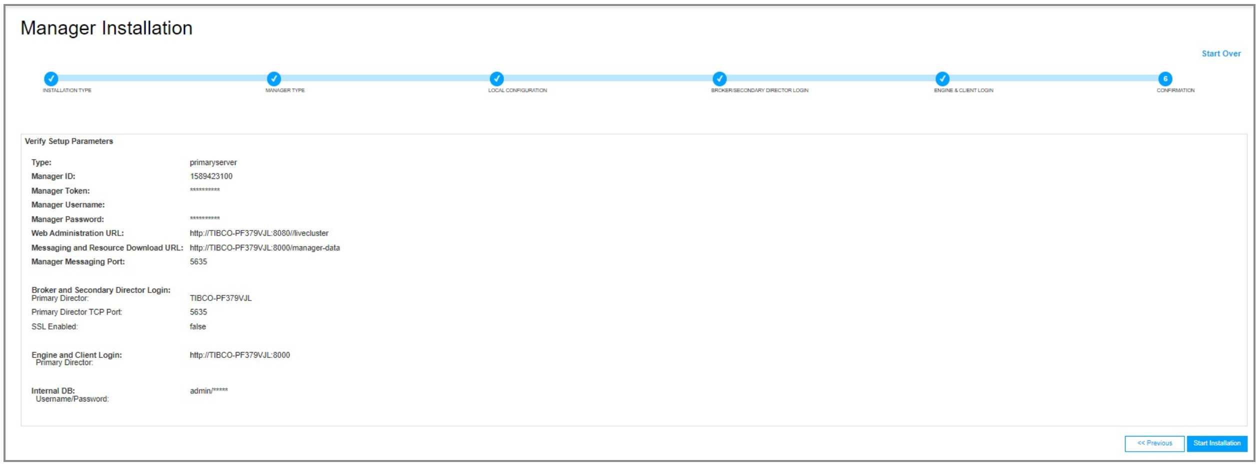
   The following issues might require different configuration than a typical install:

Issue	Configuration Information
Upgrades	Before installation, move your previous version of GridServer to another location. On the GridServer Manager Setup page, type the path to your old DS_DATA directory in the **Previous Data Directory** box. GridServer migrates your old settings to your new installation.

Issue	Configuration Information
	When upgrading from a previous version of GridServer, see Upgrading GridServer and complete all steps required before beginning this installation process.
Installing a Secondary Director or Standalone Broker	Use the two lists on the **Manager type** page to select if you want to create a Manager with no Broker, a Manager with a secondary Director, or a standalone Broker. If you install a secondary Director, reconfigure the Manager containing the primary Director. To do so, go to **Admin > System Admin > Manager Reinstall** and enter the secondary Director's address and port in the corresponding page. This configures the primary Director to recognize the secondary Director and reconfigures the Engine and Driver accordingly.

5. Verify and Install.

   After you complete a **Custom** or **Typical** installation, the **Verify Setup Parameters** screen shows the installation parameters. Review parameters to ensure they are correct. To change parameters, restart the installation process with the **Start Installation** button.



Verify Setup Parameters

Type:	primaryserver
Manager ID:	1589423100
Manager Token:	**********
Manager Username:	
Manager Password:	**********
Web Administration URL:	http://TIBCO-PF379VJL:8080//livecluster
Messaging and Resource Download URL:	http://TIBCO-PF379VJL:8000/manager-data
Manager Messaging Port:	5635
Broker and Secondary Director Login: Primary Director:	TIBCO-PF379VJL
Primary Director TCP Port:	5635
SSL Enabled:	false
Engine and Client Login: Primary Director:	http://TIBCO-PF379VJL:8000
Internal DB: Username/Password:	admin/*****

The Manager Installation screen

When installing the GridServer Manager on a UNIX system, if the installer shows the fully qualified hostname as the Manager name or uses the machine hostname as the full name, this can cause potential routing problems after installation. Change the hostname of the machine with the Custom install option. Go back to the beginning of the installation screens and repeat the installation from Step 2, and enter the correct DNS-resolvable hostname in the **Verify Hostname** screen.

When satisfied with the parameter values, click **Start Installation** to begin installation.

6. Restart the Manager

   After Manager installation finishes, you are prompted to shut down the application server. On Windows machines, use the `server stop` command; in UNIX, use the `server.sh stop` command. Restart the Manager, using the same command used earlier.

7. Create a root account

   After restarting the application server, go to `http://`*yourhost*`:`*port* again. At the prompts, enter a username, password, and password verification for your initial admin account.

   This first account you create is the *root account*. For more information about the root account, see the *TIBCO GridServer® Administration*.

   After creating your account, the home page in the Administration Tool appears.

Note: After the server is installed, you can secure `Internal.script` and passwords stored in the `installation.properties` file at `DS_DATA\conf`. For more information, see Securing Passwords and Internal.script.

Silent Installation

The Manager also has a silent installation mode, which can be used to non-interactively install and configure the Manager for automation purposes. In silent mode, the installer does not prompt for any inputs during installation. Instead, the inputs are read from a response file that is provided as a command-line parameter. The silent installer can be used for new Manager installations, or to upgrade or reinstall Manager.

The `install.silent` file is packaged in the root directory of the TIBCO GridServer® distribution. Edit the file with information for your environment before launching the silent installation. The file includes comments that describe the installation properties you can set. You can also specify properties on the command line that you don’t want to put in the properties file (such as passwords).

If errors occur during installation, they are listed in the installation log file located at DS_MANAGER`/webapps/livecluster/WEB-INF/log/install.log`.

> **Note:**
> You can provide values to the Grid authentication properties during Silent installation for Grid component authentication. For more information, see Authentication of Grid Components.

To use the silent installer:

1. Perform the following tasks:
   a. Task A: Copy Files Before Installation
   b. Task B: Configure Server Settings
2. Using a text editor, open the `install.silent` file in the root directory of the TIBCO GridServer® distribution and edit the properties for the Manager installation.

   By default, the `acceptEULA` property is set to `false`. You must set this to `true` after reading the `EULA.txt` file in the root directory of the TIBCO GridServer® installation.

   By default, the `passwordencryption` property is set to `true`. This property is used to secure `Internal.script` and passwords stored in the `installation.properties` file at `DS_DATA\conf`. For more information, see Securing Passwords and Internal.script.
3. If you are upgrading an existing Manager, ensure that it is not running.
4. Run the `install.sh` or `install.bat` file:

   `install.sh/bat` *properties-file* `[`*var1=x var2=y ...*`]`

   The *var1...varx* are optional, and enable you to specify properties on the command line instead of in the properties file (such as passwords).

> **Note**
> For upgrades performed with the silent installer, database migration does not overwrite existing entries. If you have configured the Manager and added anything to the database, the upgrade does not overwrite the new data from the previous Manager.

Configuration Options

The following configuration options can be made to customize your Manager installation.

Changing the HTTP Ports

By default, component messaging uses HTTP port 8000, and the web-based Administration Tool uses port 8080. The Tomcat server is also configured to use port 8005 for administration/shutdown.

To change any of the HTTP ports:

1. Before installation, go to the DS_MANAGER directory and run `server prepare` (for Windows) or `server.sh prepare` (for UNIX). This copies all of the application server's volatile files to the data directory.
2. In the `DS_DATA/conf` directory, edit the `server.xml` file.
3. To change the shutdown port, replace the value of 8005 in the following line:

   ```
   <Server port="8005" shutdown="SHUTDOWN">
   ```

4. To change the port used by the Administration Tool, replace the value of 8080 in the following line:

   ```
   <Connector port="8080"
   ```

5. To change the messaging port, replace the value of 8000 in the following line:

   ```
   <Connector port="8000"
   ```

6. Start the Manager and continue installation.

You can verify your changes by opening the new Web Administration URL on the new port in your browser. For the example: `http://my-manager:9000`

A Sample UNIX rc.d Script

The following is an example of a simple startup script for the GridServer Manager running on a RedHat Linux system:

```
#!/bin/sh
# Startup script for DataSynapse Manager
#
# Source function library.
. /etc/rc.d/init.d/functions
prog="server"
INSTALLDIR=/opt/TIBCO/DataSynapse/manager
DS_DATA_DIR=/var/TIBCO/DataSynapse
export DS_DATA_DIR
JAVA_HOME=/usr/local/java
export JAVA_HOME
case "$1" in
        start)
            cd $INSTALLDIR
            ./server.sh start
            ;;
        stop)
            cd $INSTALLDIR
            ./server.sh stop
            ;;
        restart)
            cd $INSTALLDIR
            ./server.sh stop
            ./server.sh start
            ;;
        *)
            echo $"Usage: $0 {start|stop|restart}"
            exit 1
esac
exit 0
```

After creating the above file, place it in `/etc/rc.d/init.d/`. Your Linux system does not directly run scripts from this directory. Instead, a different directory within `/etc/rc.d` corresponds to each runlevel of your system. When your system enters a runlevel (for example, during system startup), each script in that runlevel's associated directory runs and receives the `start` or `stop` parameter.

Instead of creating several identical copies of your script, you can create symbolic links in each runlevel directory. Links beginning with `K` run the script with the `stop` parameter; those with `S` run the `start` parameter. The number at the start of each link dictates the order in which scripts run.

The following is an example of the links created when installing the above script. These links start the GridServer Manager at runlevels 3 and 5 and stop it at runlevels 0, 1, and 6:

```
lrwxrwxrwx    1 root     root           18 Apr  8 16:26
/etc/rc.d/rc0.d/K02datasynapse -> ../init.d/dsserver
lrwxrwxrwx    1 root     root           18 Apr  8 16:27
/etc/rc.d/rc1.d/K02datasynapse -> ../init.d/dsserver
lrwxrwxrwx    1 root     root           18 Apr  8 16:27
/etc/rc.d/rc3.d/S98datasynapse -> ../init.d/dsserver
lrwxrwxrwx    1 root     root           18 Apr  8 16:27
/etc/rc.d/rc5.d/S98datasynapse -> ../init.d/dsserver
lrwxrwxrwx    1 root     root           18 Apr  8 16:27
/etc/rc.d/rc6.d/K02datasynapse -> ../init.d/dsserver
```

A Sample Systemd Unit File

Starting in Red Hat Enterprise Linux 7, the systemd init daemon is used, instead of the upstart init daemon. This uses the concept of targets defined in a unit file.

The following is an example of a `gridserver.service` unit file for the GridServer Manager running on a RedHat 7 Linux system:

```
[Unit]
Description=Datasynapse Manager service
After=network.service NetworkManager.service NetworkManager-wait-
online.service

[Service]
Type=forking
User=root
PIDFile=/opt/datasynapse/manager-data/java.pid

Environment=DS_DATA_DIR=/opt/datasynapse/manager-data
Environment=JAVA_HOME=/usr/local/java
WorkingDirectory=/opt/datasynapse/manager
ExecStart=/opt/datasynapse/manager/server.sh start
ExecStop=/opt/datasynapse/manager/server.sh stop

[Install]
WantedBy=default.target
```

After creating the above file and naming it `gridserver.service`, place it in `/etc/systemd/system/`. Refer to the Red Hat documentation for a full understanding of your options for the placement of this file.

The above example sets `WantedBy=default.target`, which creates a "want" dependency for the `default.target` related to this service. If you are using a different target in your environment, you might want to use that target instead of the `default.target` above.

Authentication of Grid Components

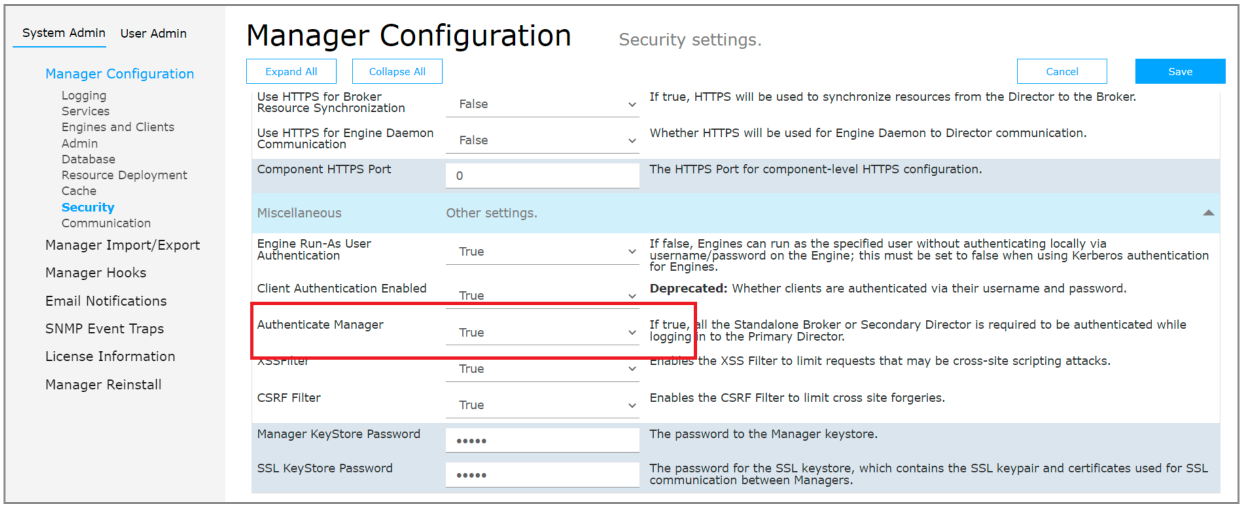
Authenticating all Grid components is an additional security option provided in GridServer. To authenticate all Grid components, you must set the following properties in the Manager:

- `Authenticate Manager`
- `Manager Token`
- `Manager UserName`
- `Manager Password`

> **Note:**
> You can also provide Grid authentication property values during silent installation for Grid component authentication.

To change the `Authenticate Manager` property, go to the **Admin > System Admin > Security > Miscellaneous** section and set the `Authenticate Manager` property to `False`.

Property	Description
`Authenticate Manager`	It is set to `True` by default. When logging in to the Primary Director, you must authenticate all Brokers or Primary Directors. If required, you can set this property on the Primary Director or on the Secondary Director.



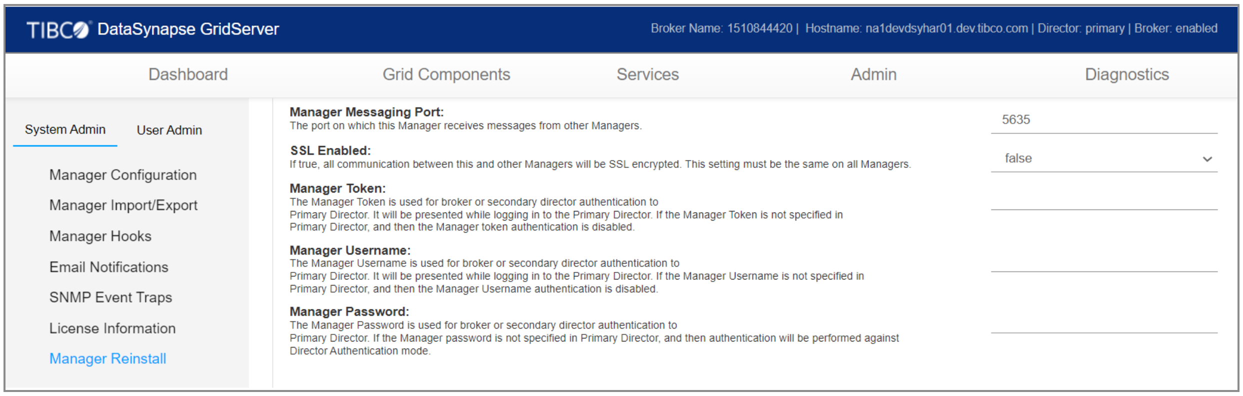
Manager Configuration Security settings.

Use HTTPS for Broker Resource Synchronization	False	If true, HTTPS will be used to synchronize resources from the Director to the Broker.
Use HTTPS for Engine Daemon Communication	False	Whether HTTPS will be used for Engine Daemon to Director communication.
Component HTTPS Port	0	The HTTPS Port for component-level HTTPS configuration.
Miscellaneous	Other settings.	
Engine Run-As User Authentication	True	If false, Engines can run as the specified user without authenticating locally via username/password on the Engine; this must be set to false when using Kerberos authentication for Engines.
Client Authentication Enabled	True	**Deprecated:** Whether clients are authenticated via their username and password.
Authenticate Manager	True	If true, all the Standalone Broker or Secondary Director is required to be authenticated while logging in to the Primary Director.
XSSFilter	True	Enables the XSS Filter to limit requests that may be cross-site scripting attacks.
CSRF Filter	True	Enables the CSRF Filter to limit cross site forgeries.
Manager KeyStore Password	•••••	The password to the Manager keystore.
SSL KeyStore Password	•••••	The password for the SSL keystore, which contains the SSL keypair and certificates used for SSL communication between Managers.

Authenticate Manager property

To configure authentication, set the following properties during installation on the Installation page:

Property	Description
`Manager Token`	It is used to authenticate the Broker or the Secondary Director when logging in to the Primary Director. If it is not specified in the Primary Director, then this token is disabled.
`Manager UserName`	It is used to authenticate the Broker or the Secondary Director when logging in to the Primary Director. If it is not specified in the Primary Director, then the Manager Username is disabled.
`Manager Password`	It is used to authenticate Broker or Secondary Director when logging in to the Primary Director. If it is not specified in the Primary Director, then authentication is performed against Director Authentication mode.



Manager Messaging Port:
The port on which this Manager receives messages from other Managers.
5635

SSL Enabled:
If true, all communication between this and other Managers will be SSL encrypted. This setting must be the same on all Managers.
false

Manager Token:
The Manager Token is used for broker or secondary director authentication to Primary Director. It will be presented while logging in to the Primary Director. If the Manager Token is not specified in Primary Director, and then the Manager token authentication is disabled.

Manager Username:
The Manager Username is used for broker or secondary director authentication to Primary Director. It will be presented while logging in to the Primary Director. If the Manager Username is not specified in Primary Director, and then the Manager Username authentication is disabled.

Manager Password:
The Manager Password is used for broker or secondary director authentication to Primary Director. If the Manager password is not specified in Primary Director, and then authentication will be performed against Director Authentication mode.

Configuring Manager Authentication

Points to note

- If you are upgrading the server and using mixed grid deployment then on the Primary Director and Secondary Director, you must set the `Authenticate Manager` property to `false`.
- Grid component authentication fails on the Secondary Director when the Primary Director goes down and the user updates the tokens on the Secondary Director (different from the Primary Director token's values) without a restart.
- The Grid components fail to log in under the following circumstance:
    - If the `Authenticate Manager` property is `true` on the Primary Director and if the token, user name, and password are not configured on the components in sync with the Primary Director.

      In this scenario, the following error is displayed on the Grid Component UI:

      `"Login is not available at this time, the Primary Director may be offline"`.

      In this case, if you do not want to use this feature, then

      1. On the Primary Director and the Secondary Director, set the `Authenticate Manager` property to `false` and restart the Primary and Secondary Director (if it exists).
      2. Restart the remaining Grid components.

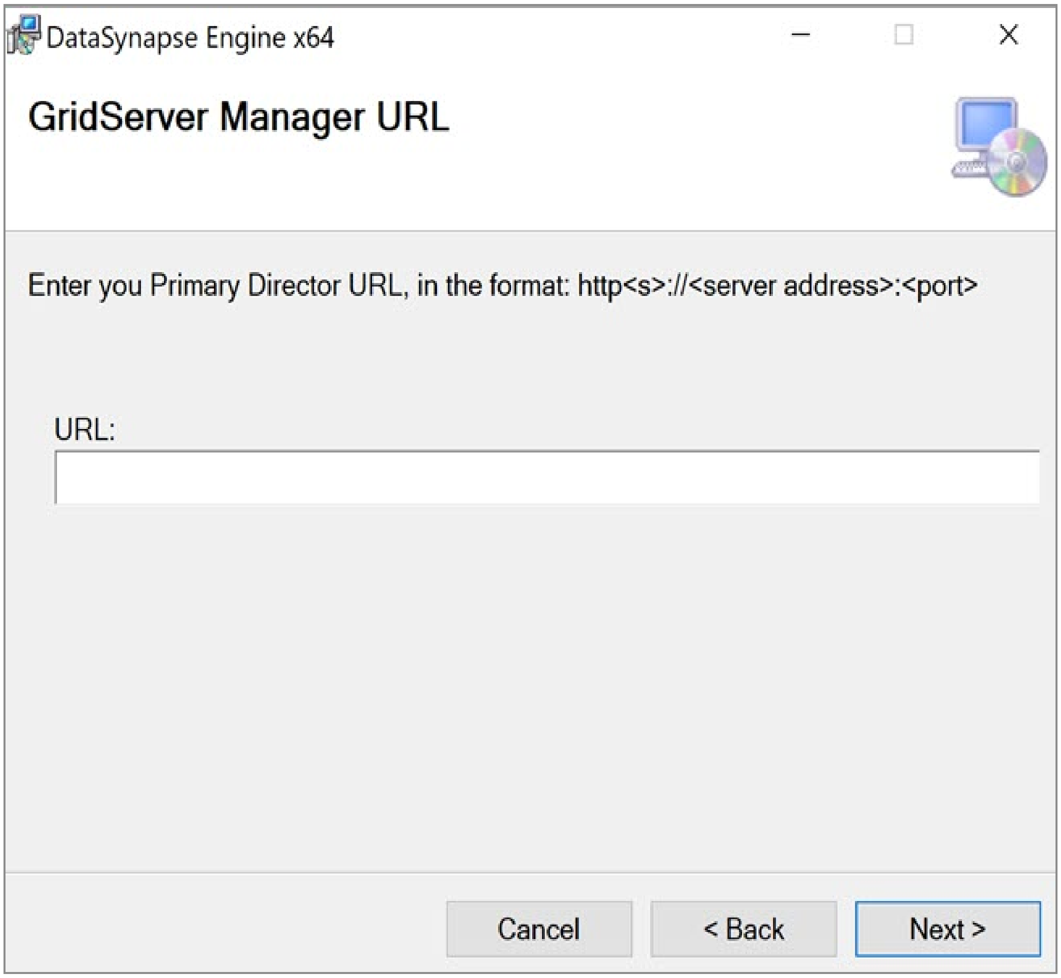
Windows Engine Installation

In the Windows environment, you can download an Engine and install it or transfer it to another computer before installation. You can also use a network installation to install Engines on many machines, or to install multiple Engine instances on different computers , or to install GridServer on a Windows Terminal Server for use by multiple PCs.

Installing Windows Engines

To install GridServer Engine on a Windows system:

1. Log in to the GridServer Administration Tool, go to the top navigation bar, and click the **Downloads** icon.
2. Click the link to download the 32-bit (x86) or 64-bit (x64) Engine installer. This downloads a file named `DSEngineInstall.msi` or `DSEngineInstall64.msi`.
3. If you are using HTTPS, you might need to copy `ssl.pem` and `ssl.keystore` to the target installation directory, see Configuring SSL.
4. Run the msi installer file, or save it and move it to another machine.
5. Click **Next** on the installer screen.
6. Select the location for the installation. This is `c:\TIBCO\DataSynapse` by default. Click **Next**.
7. Specify the URL of the Manager containing the Primary Director. This must begin with `http://` or `https://` and include the port number (8000 by default.) Click **Next**.



DataSynapse Engine x64

GridServer Manager URL

Enter you Primary Director URL, in the format: http<s>://<server address>:<port>

URL:

GridServer Manager URL

8. Enter the user name and password that must be used to run the Engine’s Windows service. Leave the fields blank to use the Local System account. Click **Next**.

Note: Using the Local System account to run the Engine service is discouraged.

9. If you want to enable Engine Daemon authentication during installation, see Enabling Engine Daemon Authentication.

10. To confirm the installation, click **Next** and then click **Close** to exit the installer.

Command Line Unattended Install

To install a Windows Engine from the command line:

1. Copy the `DSEngineInstall.msi` or `DSEngineInstall64.msi` file in `DS_MANAGER/webapps/livecluster/public_html/register/install/worker/exe` to the Engine machine or a shared directory.
2. If you are using HTTPS, you might need to copy `ssl.pem` and `ssl.keystore` to the target installation directory, see Configuring SSL.
3. If you want to enable Engine Daemon authentication during installation, see Enabling Engine Daemon Authentication.

Here is a sample command:

```
msiexec /quiet /i DSEngineInstall.msi URL=Managerhostname:port
ENGSVCUSER=service username ENGSVCPWD=service password [SSL_PEM=ssl.pem
file path] [LOGINUSER=user] [LOGINPWD=pwd] [DAEMCONFIG=config]
```

Note: Omit `ENGSVCUSER` and `ENGSVCPWD` if the Engine service should run as a Local System.

You can use the following parameters for Windows Engine installation:

Parameter	Description
`URL`	Primary Director URL
`ENGSVCUSER`	Name of the user running the Service
`ENGSVCPWD`	Password of the user running the Service
`SSL_PEM`	Optional. Path to directory that contains `ssl.pem`
`DAEMCONFIG`	Optional. Engine Daemon Configuration

Note

A custom certificate for HTTPS works only with the network installation method. If a Manager uses a custom certificate, you cannot install Engines by interactive installation.

Network Install

In most cases, executing the Windows Engine installation MSI across a network from a shared location works with no issues. If there are problems due to network speed, the contents of the MSI can be expanded into a network share and the installer runs from the network location. The MSI is expanded into a network share by performing an administrative install by adding the `/a` switch to the above `msiexec` command.

During the interactive portion of the administrative install, provide a directory into which the unpacked MSI is stored (the network share from which users install). Users then run the new `DSEngineInstall.msi` from that location (which is smaller than the original) either interactively, or by using the unattended command line parameters.

Engine JRE Synchronization

The Engine's JRE is always kept in synchronization, by default, with its Primary Director. For example, when the Engine is installed, it downloads a JRE from the Director with which it was configured. If the Engine is moved to another grid that has a different JRE, it synchronizes prior to login to that grid.

There might be certain cases in which you might not want the JRE to automatically synchronize. To disable this, add the line `jre=local` to the `intranet.dat` file. This prevents the installer from downloading and unpacking the JRE.

Note If JRE updating is disabled, it is the responsibility of the user to make sure the JRE is properly deployed on the Engines.

If an Engine's configuration uses a different Director URL than the actual name of the Director (such as one using the fully-qualified domain name and the other using a short name), the JRE is not updated when an Engine migrates. Also, if a primary Director is not available and an Engine uses a secondary Directory, the JRE is not updated.

Engine Permissions

Engines installed by an administrator but run by a non-admin user can have permission issues, especially if the Engine removes a registry key and creates it on an update event. This section contains a list of keys created and used by Engines. Use this list to help determine what permissions to set so that Engines function properly.

The following registry keys are used for performance monitoring and require only read access after install:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NetFramework
HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\CentralProcessor\0
HKEY_DYN_DATA\PerfStats\\StartStat
HKEY_DYN_DATA\PerfStats\\StatData
HKEY_DYN_DATA\PerfStats\StopStat
HKEY_LOCAL_MACHINE\SOFTWARE\microsoft\windows nt\currentversion\perflib
```

Configuring Permissions for Processor Utilization Mode

If you install an Engine Daemon to run as a Service and the Service logs in as another user, the Processor Utilization mode does not work correctly, because the Engine Daemon cannot retrieve performance data (such as CPU information) from the OS.

To remedy this:

1. Start `regedt32` (Click **Run...** from the Start menu, and type `regedt32`.)
2. Navigate to the key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Perflib`
3. Right-click the folder named `Perflib` in the left-hand pane and click **Permissions...**
4. Add the user that the Engine logs in as, and grant Read access. Do this for each Engine in the Grid.

Configuring Run-As for Windows Engines

To configure run-as for Windows Engines:

1. Right-click the Engine’s install directory, select **Properties**, and under the **Security** tab use **Add...** to add all users that you intend to run Services as.
2. Select the **Allow** check box for **Full Control**.
3. In Windows, select **Start > Control Panel > Administrative Tools > Services**. Right-click the Service running the Engine and select **Properties**. You must ensure that the Engine Daemon user can interact with the desktop. If the **Local System** user is selected, select the **Allow Service to Interact with the Desktop** check box.
4. Right-click the Engine’s `%temp%` directory, select **Properties**, and under the **Security** tab, use **Add...** to add all users that you intend to run Services as.
5. Select the **Allow** check box, for **Read**, **Write**, and **Delete** permissions. Note that the **Delete** permission is set using the **Advanced** button on the Security page of the Windows Explorer folder properties dialog box.
6. Specify Run-As credentials that have proper security privileges to run the corresponding service type for Engine instances:
User > User Admin > RunAs Credentials and then add user name and password.
7. Add Run-As user for corresponding Service Type on the UI:
Services > Service Types > Edit Service Type for corresponding Service > Add RunAs user

For more information about using Run-As, see the *TIBCO GridServer® Administration*.

Engine Configuration and Management

For more information about configuring and managing installed Engines, see the *TIBCO GridServer® Administration*.

UNIX Engine Installation

This section describes how to install GridServer Engines for UNIX. To install Engines on UNIX systems, you must download and install an Engine on each system and run a configuration script.

Installing UNIX Engines

Installing the DataSynapse GridServer UNIX Engine comprises unpacking a `gzip`-compressed `tar` archive on a UNIX system; creating or using a profile on the Manager; and associating the Engine with that profile.

To install an Engine on a UNIX system:

1. Log in to the GridServer Administration Tool, go to the top navigation bar, and click the **Downloads** icon.
2. Click the link to download the installation archive appropriate to your hardware and OS.

> ⚠ **Warning**
>
> A 64-bit UNIX Engine cannot run 32-bit native Services, such as C++ or Java with JNI.

3. Unpack the archive to the directory of your choice. For example, the following extracts the tar file into the directory `/usr/local/DSEngine`:

```
cd /usr/local
gzip -d -c DSEngineLinux.tar.gz | tar xvf -
```

4. If you are using HTTPS, you might need to copy ssl.pem and ssl.keystore to the root directory, see Configuring SSL.
5. Run the `configure.sh` script located in the directory in which you unpacked the archive.

Note Running more than one Engine from a shared network mount point is not supported.

The following table lists the `configure.sh` arguments:

configure.sh Arguments

Switch	Argument	Description
`-s`	`Manager:port`	The domain name or IP address for the Primary Director, and HTTP(S) port.
`-P`	`profiles_dir`	Optional target directory for native logging and configuration. The default value is `./profiles`
`-l`	`y|n`	Whether the Primary Director port uses HTTPS.

If you want to enable Engine Daemon authentication during installation, see the Enabling Engine Daemon Authentication topic.

Engine JRE Synchronization

The Engine's JRE is always kept in synchronization, by default, with its Primary Director. For example, when the Engine is installed, it downloads a JRE from the Director with which it was configured. If the Engine is moved to another grid that has a different JRE, it synchronizes prior to login to that grid.

There might be certain cases in which you might not want the JRE to automatically synchronize. To disable this, add the line `jre=local` to the `intranet.dat` file. This prevents the installer from downloading and unpacking the JRE.

Note If JRE updating is disabled, it is the responsibility of the user to make sure the JRE is properly deployed on the Engines.

If an Engine's configuration uses a different Director URL than the actual name of the Director (such as one using the fully-qualified domain name and the other using a short name), the JRE is not updated when an Engine migrates. Also, if a primary Director is not available and an Engine uses a secondary Directory, the JRE is not updated.

Configuring Run-As for UNIX Engines

GridServer uses engine PAM authentication service instead of system PAM authentication. Refer to your UNIX-specific PAM configuration instructions to ensure that the Run-As user is properly authenticated.

To set up the dsengine PAM service for Linux, copy the auth and account sections from the login service, dropping the secure tty module. For example:

```
login
#%PAM-1.0
auth required pam_securetty.so
auth include common-auth
auth required pam_nologin.so
auth required pam_mail.so
account include common-account
password include common-password
session include common-session
session required pam_resmgr.so
dsengine
#%PAM-1.0
auth include common-auth
auth required pam_nologin.so
account include common-account
```

For additional information, see the utilities/testpam2 section of the DataSynapse Customer Support website.

1. Start the Engine:

   ```
       ./engine.sh start
   ```

2. For Linux and Linux64, change mode of all files to be group read/writable:

   ```
       find . | xargs chmod g+u
   ```

3. Change ownership of the `invokeRA` program to `root`, and change it to be set UID:

```
sudo chown root bin/invokeRA
sudo chmod +s bin/invokeRA
```

4. Set the Engine user’s umask to make these permissions the default:

```
umask 002
```

Note that for UNIX engines, the Run-As user’s account must be in the same group as the Engine user’s account.

For more information about using Run-As, see the *TIBCO GridServer® Administration*.

Running the Engine

The `engine.sh` script follows the standard start/stop argument convention for system initialization scripts (“rc scripts”), so you can incorporate it in the start-up and shutdown sequence by inserting appropriate links in the `/etc/rc.d` files.

The `engine.sh` arguments are as follows:

engine.sh Arguments

Switch	Description
`start`	Starts the Engine
`stop`	Stops the Engine
`startfg`	Starts the Engine, and runs it in the foreground. Useful for monitoring or debugging purposes.

By default, Engine uses default configuration. To change the default configuration, use `config` argument when starting the Engine:

```
./engine.sh start -config=configValue
```

or

```
./engines.sh startfg -config=configValue
```

⚠ **Warning**

If you install a machine without a default route, the UNIX Engine returns 0.0.0.0 as the IP address. An IP address of 0.0.0.0 causes communication issues with the Manager. You must ensure that a default route is set for Engines to operate properly.

Engine Configuration and Management

For more information about configuring and managing Engines, see the *TIBCO GridServer® Administration Guide*.

Enabling Engine Daemon Authentication

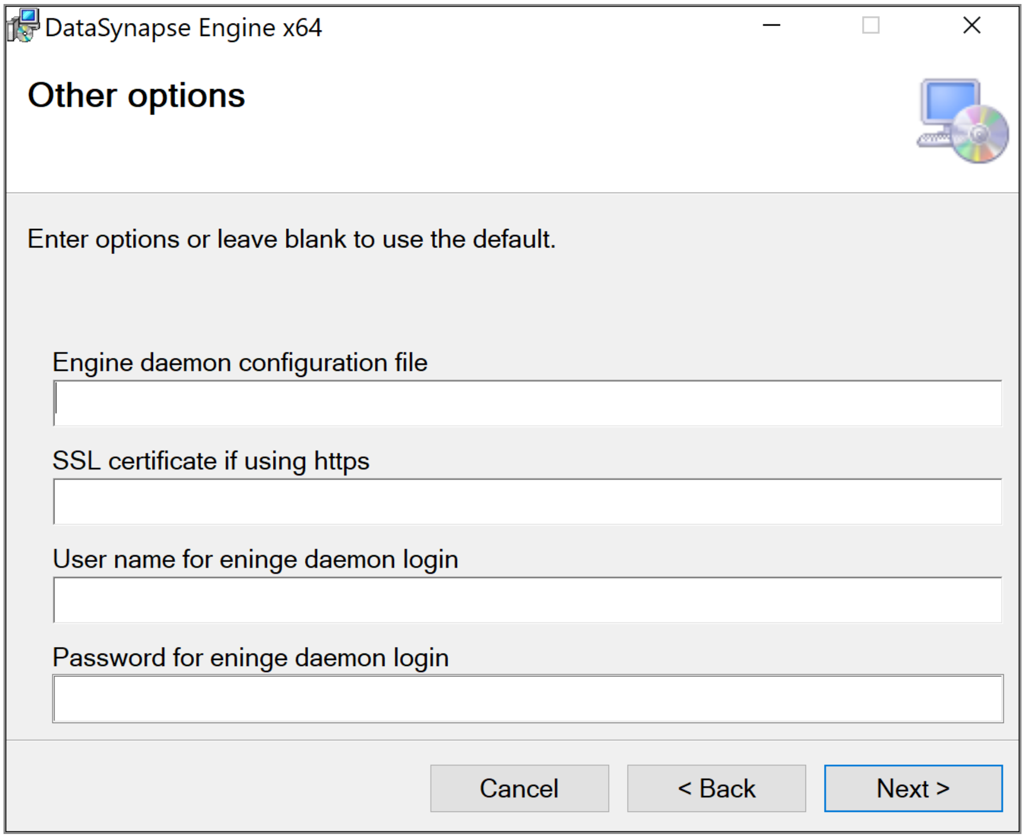
You can enable Engine Daemon authentication during Engine installation. This feature is supported only for Authentication Mode, (**Admin > User Admin > Authentication**), configured as Internal DB or LDAP.

1. Go to **Admin > System Admin > Manager Configuration > Engine and Clients** and set **Daemon Authenticate** to true.
2. Create a user with a Security Role with the **Install Engine Daemon** feature enabled. Ensure that the user has a password that is at least eight characters long.
3. During the Engine installation, depending on your operating system, perform one of the following procedures:

> **Note:** If the Engine is installed with an invalid user name or password, the Engine Daemon stops.

For Windows by using the Installer

Enter the configuration, user name, and password in the Other options dialog box.



DataSynapse Engine x64

Other options

Enter options or leave blank to use the default.

Engine daemon configuration file

SSL certificate if using https

User name for eninge daemon login

Password for eninge daemon login

Engine Daemon Authentication

For more information about installing the Windows Engine by using the Windows Engine Installer, see Installing Windows Engines.

For Windows by using the Command Line Interface

To enable Engine Daemon Authentication by using the command line, run the following command:

```
msiexec /quiet /i DSEngineInstall.msi URL=Managerhostname:port
LOGINUSER=user LOGINPWD=pwd DAEMCONFIG=config
```

Parameter	Description
LOGINUSER	User name for Engine Daemon login
LOGINPWD	Password for Engine Daemon login

For more information about installing the Windows Engine by using the Windows command line, see Command Line Unattended Install.

For UNIX

Use the following command:

```
./engine.sh start -installuser=user -installpwd=password -config=myConfig
```

The following table lists the `engine.sh` arguments:

engine.sh Arguments

Argument	Description
`installuser`	User name for Engine Daemon login
`installpwd`	Password for Engine Daemon login

For more information about installing Unix Engine, see Installing UNIX Engines .

To Reset the User Name or Password

Procedure

1. Delete the `DefaultInstallOption.dat` file present in the `Engine` directory.
2. From the command line, start the Engine based on the operating system:

- For Windows, use the following command:
  `Engine.exe start -installuser=user -installpwd=password -config=myConfig`
- For UNIX, use the following command:
  `./engine.sh start -installuser=user -installpwd=password -config=myConfig`

Alternatively, you can also perform the following steps to reset the user name or password:

Procedure

1. Open the `DefaultInstallOption.dat` file present in the `Engine` directory.
2. Delete all the content and then add the user name or password as follows:
   ```
   installuser=user
   installpwd= pwd
   ```

Verify Installation

After installing a GridServer Manager and one or more Engines, you can test your installation by submitting a test Service and watching its progress in the GridServer Administration Tool.

Submitting a Service Test

A Service Test can run a Linpack benchmark that you can use as a basic test for your Grid.

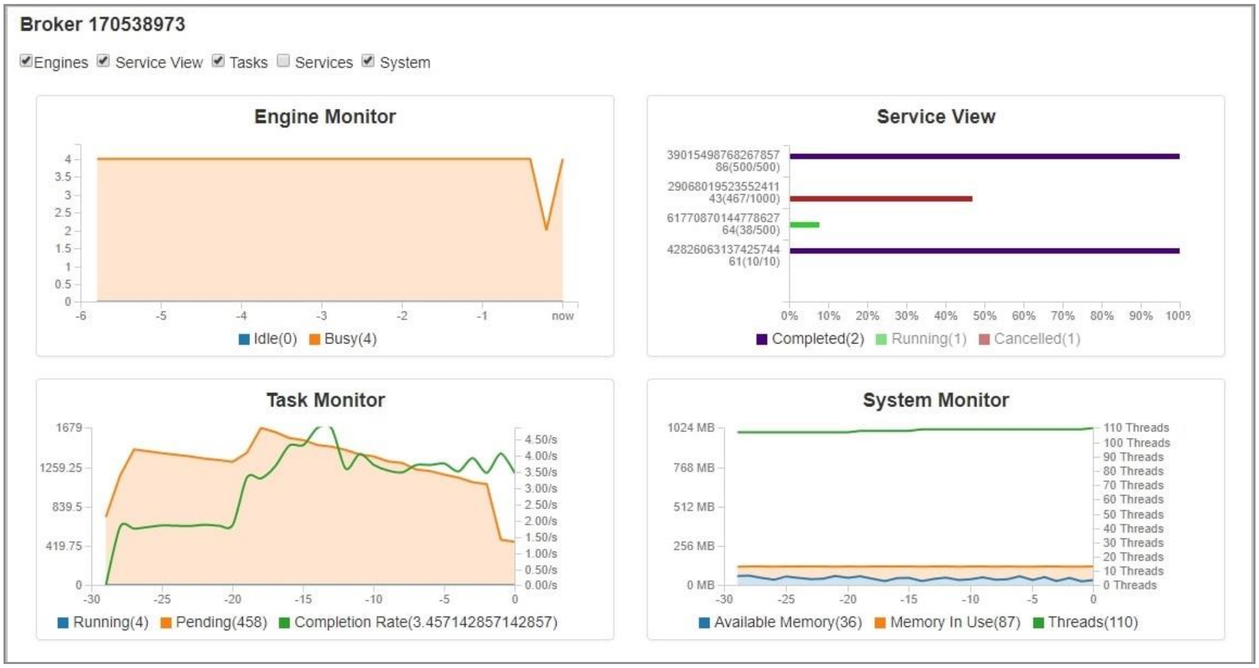
To submit a Service Test:

1. Make sure at least one Engine is logged in to your Manager.
2. Log in to the GridServer Administration Tool and go to **Services > Services > Service Test**.
3. Click **Submit** to send the Linpack Test with default values. You can also change the number of Tasks or Duration to make a Service that takes longer to complete.
4. The **Services > Services > Service Session Admin** page appears. From this page, you can monitor the Service's progress.

Monitoring a Service Test

After you submit the Service Test, you can monitor its progress to ensure that your Engine is taking tasks and returning results. While the Service Test is running, you can also view other pages for more information:

- Go to **Grid Components > Brokers > Broker Admin**. A list of available Brokers appears, along with the number of busy, idle, and total Engines reporting to that Broker.
- Go to **Dashboard > Broker Monitor**. A screen appears showing four interactive graphs: Engines, Service View, Tasks, and System. The System Monitor graph shows memory and thread activity.



The Broker Monitor

- Go to **Grid Components > Engines > Engine Admin** for a list of all Engines and their status.

Troubleshooting

If the Service Test does not function, or you have other issues relating to your installation, consult the *TIBCO GridServer® Administration* and online help in the Administration Tool.

Uninstalling GridServer

This section describes how to uninstall GridServer. The uninstall procedures for the Windows operating system and the UNIX operating system differ.

Uninstalling GridServer on Windows Systems

Uninstalling an Engine

To uninstall an Engine on a Windows system, first ensure that the Engine is stopped and then go to **Start > Control Panel > Add/Remove Programs** or **Start > Control Panel > Programs and Features**. Remove the **DataSynapse Engine**. Follow the onscreen instructions through the uninstall.

Manual Uninstall

Uninstall can be done from the command line as follows:

```
msiexec /x DSEngineInstall.msi
```

or

```
msiexec /quiet /x DSEngineInstall.msi
```

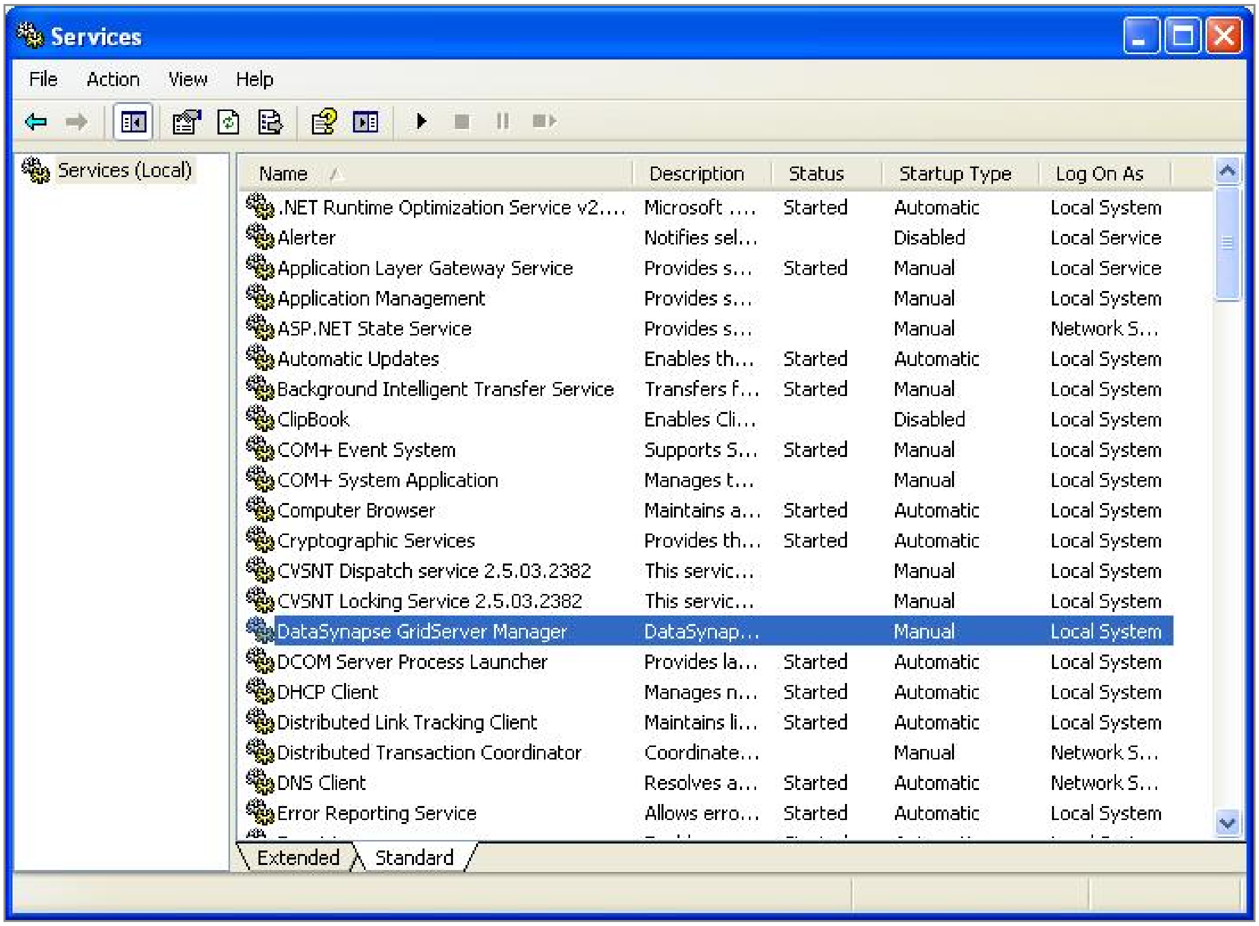
Instead of passing the name of the MSI file to `msiexec`, you can alternately substitute the Product Code, which is `{F80D0E61-23AF-4934-A40B-190F2A1C52CE}` for both 32-bit and 64-bit Engines. For example:

```
msiexec /x {F80D0E61-23AF-4934-A40B-190F2A1C52CE}
```

Uninstalling a Manager

To uninstall a Manager installed as a service on a Windows system:

1. In Windows, go to **Start > Control Panel > Administrative Tools > Services**.
2. Click the **Standard** tab.
3. Click **DataSynapse Manager** and click the Stop icon to stop the application.



Name	Description	Status	Startup Type	Log On As
.NET Runtime Optimization Service v2....	Microsoft	Started	Automatic	Local System
Alerter	Notifies sel...		Disabled	Local Service
Application Layer Gateway Service	Provides s...	Started	Manual	Local Service
Application Management	Provides s...		Manual	Local System
ASP.NET State Service	Provides s...		Manual	Network S...
Automatic Updates	Enables th...	Started	Automatic	Local System
Background Intelligent Transfer Service	Transfers f...	Started	Manual	Local System
ClipBook	Enables Cli...		Disabled	Local System
COM+ Event System	Supports S...	Started	Manual	Local System
COM+ System Application	Manages t...		Manual	Local System
Computer Browser	Maintains a...	Started	Automatic	Local System
Cryptographic Services	Provides th...	Started	Automatic	Local System
CVSNT Dispatch service 2.5.03.2382	This servic...		Manual	Local System
CVSNT Locking Service 2.5.03.2382	This servic...		Manual	Local System
DataSynapse GridServer Manager	DataSynap...		Manual	Local System
DCOM Server Process Launcher	Provides la...	Started	Automatic	Local System
DHCP Client	Manages n...	Started	Automatic	Local System
Distributed Link Tracking Client	Maintains li...	Started	Automatic	Local System
Distributed Transaction Coordinator	Coordinate...		Manual	Network S...
DNS Client	Resolves a...	Started	Automatic	Network S...
Error Reporting Service	Allows erro...	Started	Automatic	Local System

Stopping a Manager installed as a Windows Service

4. Open a command line, go to the DS_MANAGER directory, and issue a `service remove` command to uninstall the service.
5. Remove the DS_INSTALL directory and its contents.

Uninstalling GridServer on UNIX Systems

To remove a GridServer Manager or Engines on a UNIX system:

1. Stop the Engine or the Broker (`engine.sh stop` or `server.sh stop`).
2. Remove any modifications you made to startup files, typically located in `/etc/rc.d` directories.
3. Remove the DS_INSTALL directory and its contents.

Upgrading GridServer

This section explains how to upgrade GridServer components. It contains a GridServer Upgrade Checklist, which explains what to do before, during, and after your new GridServer installation to ensure carrying over old values to the new installation, when possible.

Requirements Changes

See the system requirements in the TIBCO GridServer® readme file to ensure that your software and hardware meet the minimum system requirements. Ensure to review the full list of requirements.

GridServer 7.1.0 Upgrade Checklist

You can migrate settings from your previous version of GridServer to your new version of a GridServer installation, but it's important to not install on top of your previous installation. Instead, shut down your old Manager, and rename the old installation directory. You can then install the new version of GridServer in the same location. Later in the install, you supply a path to the renamed old installation so settings can be copied. An `upgrade.log` file is generated that lists all changed configuration values, as well as settings that could not be migrated to the new installation.

The only direct upgrade supported on GridServer version 7.1.0 is from GridServer versions 7.0.x and 6.3.1. To upgrade from any other earlier version to GridServer 7.1.0, you must first upgrade to GridServer version 6.3.1.

Note:
Authenticating all Grid Components is an additional security option provided in GridServer. For more information, see Authentication of Grid Components

Prior to Upgrade

- If you have any UNIX Engines that share a common installation directory, you must migrate those installations to a local directory. A shared install point is no longer supported.
- To enable programmatic version control, ensure that `gridLibraryStrictVersioning` is set to `false`.
- If you have a Linux64 Engine, for upgrade from all versions earlier than 7.0.0_hotfix08, you must perform the following steps:
  a. Stop the server.
  b. As a best practice, take a backup of the existing `engine.sh` file.
  c. Replace the `engine.sh` file from the `engineScriptUpdate` folder to `datasynapse/manager/webapps/livecluster/engineUpdate/linux64/ and datasynapse/manager-data/engineUpdate/linux64/` folder.
  d. Start the server.
  e. Let the Engines log in and then restart the Engine Daemon.
  f. Stop the server.
  g. Upgrade the server.

> **Note:** In case of Grid Architecture, perform the steps from step a to step g on the Primary Director.

- Check the list of changes in the release notes for any other issues that might affect you.

After Stopping the Manager

- When running GridServer as a Windows service, remove the service prior to upgrading.

The following steps are necessary only if you set up the system by using an external database:

- Copy the corresponding database properties file from old `DS_MANAGER/webapps/livecluster/WEB-INF/etc/db/` directory to the current installation. This is used when you run the database upgrade script.
- Upgrade your external reporting database schema using the `upgradedb.sh` or `upgradedb.bat` script located in `DS_MANAGER/webapps/livecluster/WEB-INF/etc/db`.

During Installation

- For **Installation Type**, choose **Custom, Manager Upgrade**.
- For **Previous Data Directory**, enter the pathname of your previous base directory.

After Installation

If Using System Classloader for Engines

If you are using the system classloader in your Engine configuration and deployed the `DSEngine.jar` file to Engines, deploy the new `DSEngine.jar` to Engines after upgrade. This is now deployed in `DS_MANAGER/webapps/livecluster/engineUpdate/shared`. If `DSEngine.jar` was deployed in the default resources directory, it has already been removed. If it was deployed to any Grid Libraries, you must remove it from them.

External Database JDBC Drivers

If you are using an external database, copy the required JDBC driver JAR archive from `DS_MANAGER/webapps/livecluster/WEB-INF/lib` in the old installation to `DS_MANAGER/webapps/livecluster/WEB-INF/lib/` in the new installation.

LDAP Upgrade

As of GridServer 6.0, Driver profiles are no longer supported. The `allowedBrokers` property has been replaced by the value of **Manager List** in the security role assigned to the Driver user. Also, users are now allowed to execute Services based on the “Execute Services” permission rather than by virtue of having a profile.

During upgrade, a new role is created for each Driver profile. This role has execute permission, and maps the “Allowed Brokers” field to “Manager List.” That role is added to each user who has the corresponding profile.

You are not required to maintain these converted profiles; you might choose to reorganize and remap these roles as you see fit.

If you use LDAP authentication and you prefer to assign roles via LDAP groups, you must add those groups to your users in LDAP, and remove the locally assigned roles.

Users and roles are logged during the upgrade to the file `DS_DATA/logs/newRoles.txt`, so that you have a list of groups to add the user’s LDAP entries. It is a flat file with the following format:

```
[user]\t[role1,role2,...]
```

For example:

```
joe driver_developers,driver_boston
tom driver_testing,driver_london
```

Engine Upgrade

- For UNIX Engines, `bin/invokeRA` cannot be automatically upgraded. After upgrading, you must manually copy the new `bin/invokeRA` to the upgraded Engine, and follow the instructions in “Using Run-As” in the *TIBCO GridServer® Administration*.

Driver Upgrade

- Download the new version of the GridServer SDK in the GridServer Administration Tool.
- Make sure that all Drivers are upgraded. This simply involves replacing the libraries or executables with the new version.
- If you have existing CPPDriver applications, relink them with the new libraries included in the SDK.
- On the Driver side, link your code with the `dsUtil` library. Link Engine-side Service code with the `dsUtil` library if it uses any exception classes.

Note:
After the server is upgraded, you can secure the `Internal.script` and passwords stored in the `installation.properties` file at `DS_DATA\conf`. For more information, see Securing Passwords and Internal.script in the *TIBCO GridServer® Manager Installation Guide*.

Mixed-Version Deployments

Mixed-version deployments are grids that consist of GridServer Manager versions 7.1.0, 7.0.x, and 6.3.1. This is an allowed configuration, as a convenience for large Grids that cannot upgrade all Brokers at once. You must address the following issues when running mixed-version deployments:

- The Directors must be upgraded first to the latest version of GridServer Manager 7.1.0.
- All Brokers must be on the latest service pack and hotfix of the minor version.
- You must disable client version checking for Engines, Drivers, and Brokers at **Admin > System Admin > Manager Configuration > Admin**.
- Hard-partitioning of Engines is required. That is, Engines must not be allowed on all the three GridServer Manager versions 7.1.0, 7.0.x and 6.3.1 Brokers. This is done with the balancer configuration.
- You must deploy all versions of the C++ Bridge Grid Libraries via the Primary Director. If you upgraded the Director, the older bridges might already be there, but if the Director is a new installation you must copy them from your old Managers. The Engines automatically choose the proper library based on the Broker version. Because both versions of the C++ Bridges are used, Grid Libraries must also be used on an older Broker if the code is native, rather than using the standard resource deployment.

Mixed-Version Driver Compatibility

GridServer Manager 7.1.0 has backward compatibility with GridServer Manager 7.0.x and 6.3.1 Drivers.

Rolling Upgrades

Use the following procedures to install an update to a large grid (a grid with a large number of Engines and several Brokers) in situations where you cannot permit grid-wide Engine restarts and interruptions in Service availability. These instructions enable Services already deployed on components using a previous version of GridServer Manager to continue running as other components on the grid are updated.

Prerequisite Actions

For all Managers:

1. Bring all Managers up to the required hotfix level.
2. Confirm that all GridServer Managers are at that level. On the Primary Director, you can get the versions of all Brokers on the **Grid Components > Brokers > Broker Admin** page. To find the version of the Directors, click the **About** link at the top of the Administration Tool.
3. In each Engine Configuration, confirm that **Log off Engines on Daemon Logoff** is set to false (the default setting). This prevents unnecessary Engine restarts when the Director is restarted.
4. On each Broker, go to **Admin > System Admin > Manager Configuration > Admin** and set **Daemon Upgrade From Broker** to false; then go to **Admin > System Admin > Manager Configuration > Resource Deployment > Broker Settings** and set **Synchronize Resources From Director** to false. This prevents all Daemons from restarting to upgrade when the Primary Director is upgraded.
5. On each Director, go to **Admin > System Admin > Manager Configuration > Admin**. Confirm that **Daemon Upgrade From Director** is false. (This is the default and would rarely be true.)
6. To assist in the event of any issues, set the logging levels on all Managers and Engines to Fine.

Upgrading a Grid

Note that each Manager upgrade procedure does not make any modifications to the previous installation. If required, you can use this directory to roll back the upgrade.

Primary Director Upgrade

1. Shut down the Primary Director. The Engine Daemons move to the Secondary Director and the Engine Instances (which might be running tasks) continue without interruption.
2. Upgrade the Primary Director.
3. Start the Primary Director.
4. Verify the the following Director and Broker mismatch controls are True: **Admin > System Admin > Manager Configuration > Admin > Version Management >Allow Director Version Mismatch** and **Admin > System Admin > Manager Configuration > Admin > Version Management >Allow Broker Version Mismatch**.
5. Check that all the Brokers are logged in to it at **Grid Components > Brokers > Broker Admin**. The Engine Daemons move to the Primary Director. The Engine Instances do not restart.

Secondary Director Upgrade

Follow the above procedure for the Primary Director. Shut down the Secondary Director and upgrade.

Broker Upgrades

Failover Brokers must be upgraded after regular Brokers.

For each Broker:

Procedure

1. Shut down the Broker. Within a few minutes Engines and Drivers migrate to a Failover Broker.
2. Upgrade the Broker.
3. Start the Broker. Verify that it logs in to the Directors.

4. Verify that **Admin > System Admin > Manager Configuration > Admin > Version Management > Allow Driver Version Mismatch** is True.
5. Within a few minutes Engines and Drivers migrate back.

Upgrade the Engine Daemons

At this point, the Grid has been updated, but Engine Daemons are still at the previous version. Daemon upgrades result in Engine restarts, so do the following on each Broker when the time is appropriate:

- On each Broker, go to **Admin > System Admin > Manager Configuration > Admin**. Set **Daemon Upgrade From Broker** to true.

Note The above step needs to be done as soon as possible.

Installing Hotfixes

To install or uninstall a hotfix, refer to the installation instructions in the readme file distributed with the hotfix.

Securing Passwords and Internal.script

You can enable or disable the encryption of the internal database password and the reporting database password stored in the `installation.properties` file. You can also make the `Internal.script` file available in an unreadable form.

The Grid Server can be installed or upgraded in the GUI installation mode and silent installation mode. The procedure to enable encryption for these installation modes is explained later in this section.

Considerations for Enabling Encryption

- **For the Internal.script file:** This change is only for Primary Director and Secondary Director because there is no separate internal database for Brokers. If you set the property to `true` on the Primary Director and `false` on the Secondary Director, the `Internal.script` on the Secondary Director becomes unreadable. This happens because the Secondary Director backs up the database from the Primary Director.
- **For mixed Grid Deployment:** In case the property is set to true, mixed deployment of the Grid is not supported. Ensure that all components use version 7.1.0 of GridServer Manager.

Enable or Disable Encryption (GUI Installation Mode)

1. Install or upgrade the server as required.
2. Go to the *DS_DATA*`\conf` folder and open the `installation.properties` file for editing.
3. In the `installation.properties` file, set the `DSPasswordEncryption` property to `true` (to enable encryption) or `false` (to disable encryption) as required.
4. Reinstall the server from **Admin > Manager Reinstall**.

Enable or Disable Encryption (Silent Installation Mode)

1. Go to the *DS_MANAGER* folder and open the `install.silent` file for editing.
2. In the `install.silent` file, set the `passwordencryption` property value to `true` (to enable encryption) or `false` (to disable encryption) as required.
3. Install or upgrade the server as required.

ℹ **Note:** To change the property later, perform the steps mentioned in Enable or Disable Encryption (GUI Installation Mode).

Configuring SSL

This section provides information about configuring SSL to provide secure communication between GridServer components.

About SSL

SSL (Secure Socket Layer) communication can be enabled for communication at each level in the GridServer architecture depending on the security requirements of the organization and the deployment scenarios involved. SSL provides both encryption of messaging between components, and a trust relationship of the server by the client. SSL over sockets can be used for Manager inter-communications, and HTTPS (HTTP over SSL) can be used for all other components and for resource synchronization.

Communication Overview

To understand how SSL is used for messaging, it is important to understand how components establish communication channels with each other. For the remainder of this discussion, the terms “client” and “server” are used in the traditional way, that is, a client/server relationship. An example is the Engine Daemon is a “client” to the Director’s “server”.

There are two aspects to establishing communication. The first step is the login process. The client requests a login through a known communication channel. At that point, the server might perform authentication or validation, and if successful, it returns a connection for use from then on. Note that this channel might be on a different server. For example, an Engine logs in to a Director, but the connection exists on a Broker.

SSL is configurable for both aspects. If SSL is to be used for login, it must be configured on the client. If SSL is to be used for the connection, it must be enabled on the server. For example, to enable a Driver to login using SSL, the Driver must be set to the HTTPS URL address on the Director, either in the `driver.properties` file or with the API. To enable HTTPS communication between the Driver and Broker after login, it must be set on the Broker, typically by configuring all Messaging and Download URLs to the HTTPS URL.

Certificate Overview

All SSL clients establish a trust relationship with their server. This is performed with a certificate on the client side, which is a public key that is associated with a private key on the server. When establishing the trust relationship, the server's certificate must either have been signed by a key trusted by the client, or be trusted explicitly by the client (a self-signed certificate). Most SSL clients contain a set of trusted Certificate Authorities (CAs), so that if a server has a certificate signed by one of those CAs, it automatically trusts the server. If the server is self-signed, that server's certificate must be added to the client's list of trusted servers.

In addition, the client checks the Common Name (CN) of the server's certificate against the hostname of the server, to verify that the certificate is being used on the intended host. This means that all Servers require a certificate chained to the top-level certificate.

For Java clients, such as Java Drivers and Engines, the certs are kept in a trust keystore. The JRE is shipped with a default keystore called *cacerts*. To add a certificate, typically this file is copied, the certificate is imported, and the JVM is pointed to the new file.

Native clients that use the cURL library use a platform-dependent directory that contains CA certificates. GridServer Engine Daemons and C++ Drivers also look for a PEM-encoded certificate called `ssl.pem` in the current working directory.

.NET clients look in the Windows Trust Store local to that machine. Certificates can be added to that store by double-clicking on the file and importing using the wizard.

For demonstration purposes, GridServer is packaged with a demo self-signed key-pair and certificate. All clients have a local copy of the certificate added to their list of trusted servers. While this self-signed cert is useful to demonstrate how to set up SSL, a certificate for your organization must be used in a production environment.

Key and Certificate Location and Generation

The following is a list of all key and certificate files used by the Servers and Clients.

Key and Certificate Files

File	Locations	Description
`server.key`	`DS_DATA/certs`	The private key used by OpenSSL on Tomcat for HTTPS.
`server.crt`	`DS_DATA/certs`	The certificate used by OpenSSL on Tomcat for HTTPS.
`ssl.pem`	• `DS_DATA/engineUpdate/shared` • In the current working directory of the C++ Driver	Used by native clients when self-signed.
`server.keystore`	`DS_DATA/certs`	A keystore that contains the `server.key`, used by Manager Socket SSL.
`ssl.keystore`	• `DS_DATA/engineUpdate/shared` • `DS_DATA/certs` • In the working directory or classpath of the Java Driver	A keystore that contains the root certificate and all CA certs, used by Manager Socket SSL, Java Drivers, and Engines, when self-signed.
`manager.keystore`	`DS_DATA/certs`	The keystore used by the Manager for encrypting data, such as user passwords. It is not used for SSL.

Manager HTTPS Configuration

The HTTP and HTTPS connections are configured in the Tomcat configuration file, `DS_DATA/conf/server.xml`. By default, there are four connector elements: two HTTP elements and two commented out HTTPS elements. On a Manager Install or Reinstall, this file is read; the first element is used as the Administration Tool connector, and the second is the messaging connection. These connectors are mapped to a Manager as follows:

- Administration Tool Connector: Any time a user attempts to log in over any other connector, the user is redirected to this connector.
- Messaging Connector: The Director URL is set to use this connector, as are the connections established after log in.

Configuring HTTPS

The following procedure is used to configure HTTPS:

Task A Prepare Certificates

Procedure

1. Create a temporary directory (such as `/opt/mycerts`) and execute the following commands within it.
2. Copy the `cacerts` file from the JDK to the file `ssl.keystore`.

   For JDK8:

   ```
   cp $JAVA_HOME/jre/lib/security/cacerts ssl.keystore
   ```

   For JDK11:

   ```
   cp $JAVA_HOME/lib/security/cacerts ssl.keystore
   ```

3. If you have a signed certificate:

   a. If your certificate is from a well-known CA whose root certificate is already in `$JAVA_HOME/jre/lib/security/cacerts`, it is in `ssl.keystore`.

      If it is not, add your CA's root certificate to the `ssl.keystore`:

      ```
      keytool -importcert -file CA_ROOT.crt -keystore \
      ssl.keystore -alias CA_ROOT
      ```

b. If you have a signed certificate `.pem` file and the private `.key` file that was used to generate the certificate request, copy or rename them to `server.crt` and `server.key` respectively.

4. If you don’t have a CA signed certificate and need a self-signed certificate, generate a self-signed certificate (and its associated private key):

```
     openssl req -x509 -sha256 -nodes -days 1826 -newkey \
       rsa:2048 -keyout server.key -out server.crt
```

5. If you are using a self-signed certificate, import the new certificate into `ssl.keystore`:

```
     keytool -importcert -file server.crt -keystore \
     ssl.keystore -alias MySelfSigned -storepass changeit
```

6. Copy the `server.crt` file to `ssl.pem`:

```
    cp server.crt ssl.pem
```

7. Create a PKCS #12 format file that contains your keypair:

```
openssl pkcs12 -export -in server.crt -inkey server.key -out
server.pkcs12
```

8. Convert it to a JKS format keystore:

```
keytool -importkeystore -srckeystore server.pkcs12 -srcstoretype
pkcs12 -srcstorepass changeit -destkeystore server.keystore -
deststoretype jks -deststorepass changeit
```

9. Copy the files into your Manager installation:

```
    cp ssl.keystore ssl.pem DS_DATA/engineUpdate/shared
    cp server.crt server.key ssl.keystore server.keystore \
       DS_DATA/certs
```

10. If you are enabling SSL on both a standalone Director and standalone Broker (both installed on different hosts), the previous step must be followed for each individual Manager installation. Also, while enabling SSL on a standalone Broker, after following the steps listed above, the standalone Director’s `server.crt` must also be imported to the `ssl.keystore` of the standalone Broker:

```
keytool -importcert -file dserver.crt -keystore ssl.keystore \
  -alias MySelfSignedDir -storepass changeit
```

Here, the `dserver.crt` is the `server.crt` from the standalone Director.

11. Copy the latest `ssl.keystore` and `ssl.pem` files on standalone Director and Broker's directory.

```
cat dserver.crt server.crt<Broker server.crt>  > ssl.pem
```

Task B Enable HTTPS/SSL on the Manager

Procedure

1. Edit the `DS_DATA/conf/server.xml` file:

   a. Comment out or remove the two connector elements for ports 8000 and 8080.

   b. Uncomment the two connectors for ports 8443 and 8043.

   c. Add `SSLPassword="changeit"` (or your keystore password value) to the 8043 and 8443 connectors.

Note If you plan to change your keystore password from the default value of `"changeit"` to some other value, then follow the steps given at Optional Task: Change the Default Value Keystore Password.

2. If already installed, stop your Manager, open the `DS_DATA/conf/installation.properties` file, and set `DSConfigureOnStartup=true`.

3. Start the Manager, and perform a Manager Install. On the **Local Configuration** step, you must see your new HTTPS values. Make sure to enable SSL. Complete the install and restart the Manager.

4. If you have already installed Drivers from this GridServer installation, you must edit their `driver.properties` files to use the new HTTPS URL before they use SSL. Engines reconfigure themselves to use the new secure reinstallation; the Director URLs in all Engine Configurations change to `https://host:sslport`.

Task C Configure the Manager

Procedure

1. In the Administration Tool, go to **Admin > System Admin > Manager Configuration > Security**.
2. You can set any or all of the SSL parameters on this page to true.
3. To configure the keystore passwords, change the passwords under the **Miscellaneous** heading.
4. Under **HTTPS Communication**, set **SSL Port** to 8043.
5. Restart the Manager.

Task D Configure the Clients

Procedure

1. If you already installed Drivers and Engines from this TIBCO GridServer® installation, you must change their properties files to point to the new HTTPS URL before they use SSL. Configure the following:
   - Drivers: Driver HTTPS
   - Engines and Engine Daemons: Engines and Engine Daemon HTTPS

Optional Task: Change the Default Value Keystore Password

You must do the following changes before you make changes in the certificate and restart the Manager:

- Change the value in the Administration Tool at **Admin > System Admin > Manager Configuration > Security > Miscellaneous > Manager Keystore Password** (for `manager.keystore`) and **SSL Keystore Password** (for `server.keystore`).
- In the `server.xml` file, add the `keystorePass="yournewpassword"` parameter to the 8043 and 8443 connectors.
- When reconfiguring the Manager (Primary Director and Broker, or Primary Director) to SSL Broker (Standalone or Failover), under the `DS_DATA/conf` location, you must make the following additional changes:

   a. Rename `director.properties` to `broker.properties`.

b. Edit `broker.properties` and make the following changes:

— Replace `"/messageserver[@name\="Director"]"` with `"/messageserver [@name\="Broker"]"`

— Remove the following line, if it exists:

```
/messageserver[@name\="Director"]/plugin
[@class\="DirectorInfoPlugin"]/property
[@name\="AuthenticateManager"]=False
```

Driver HTTPS

The following files are used for Driver HTTPS:

- JDriver: `ssl.keystore` — Includes the self-signed DataSynapse certificate plus CA certs. This is not needed if you are using a CA-signed certificate. The Driver looks for this in the same directory as `driver.properties`, or you can set this location with the API. It can also be placed in the classpath or working directory.
- .NET Driver: `ssl.crt` — This is `ssl.pem` renamed so that Windows recognizes it. This is not needed if you are using a CA-signed certificate. If you do, double-click on the cert to install it.
- C++ Driver: `ssl.pem` — On a UNIX machine, it might not be needed if using a CA-signed certificate, since most distributions place CA certs in locations known to OpenSSL. It is required by Windows. The Driver looks for this in the working directory.
- Python Driver: Copy the updated `ssl.pem` file to the `./GridServerSDK-Linux/examples/service/python/client` folder.

Engines and Engine Daemon HTTPS

To enable Engines and Engine Daemons to trust the Broker:

1. Place the `ssl.pem` and `ssl.keystore` files in the DS_DATA`/engineUpdate/shared` directory.
2. Copy both files to the root directory of each Engine installation. You must copy the `ssl.pem` and `ssl.keystore` files by hand to each Engine already installed, and to any subsequently installed Engines.

3. When installing Engines, you must use the SSL port, 8043 and for Windows use `https` in the URL. For UNIX, you must give the `-l y` arguments to use SSL.

You must enable HTTPS on the Manager for login, connection for Engine Daemons, and connection for Engine instances.

To enable SSL for Engine instance and Engine Daemon login, you must set the Managers to the HTTPS location in the Engine Configuration.

To enable SSL for Engine communication, you must enable it on the Manager. SSL is enabled for Engine Daemons on Directors. If your Manager is configured to use HTTPS for all Messaging, Engines already use HTTPS.

To enable SSL for Engines if you did not enable HTTPS for all messaging:

Procedure

1. Go to **Admin > System Admin > Manager Configuration > Security**.
2. Under **Component HTTPS Communication**, set **Use HTTPS for Engine Communication** to True.
3. Click **Save**.

To enable SSL for Engines Daemons on a Manager:

Procedure

1. Go to **Admin > System Admin > Manager Configuration > Security**.
2. Under **Component HTTPS Communication**, set **Use HTTPS for Engine Daemon Communication** to True.
3. Click **Save**.

To enable domain-only authentication, you can use a wildcard (*) for the hostname in the CN field of your certificate. (For example. your CN could be set to `*.example.com`.)

Broker and Director SSL

The communication between Manager components (such as the connection between Brokers and Directors, and the connection between Primary Director and Secondary Director) can be configured to use SSL. Note that because they use pure sockets for communication, HTTPS does not need to be enabled on the Manager.

If you are enabling SSL for Manager components, it must be enabled on **all** Managers.

⚠ **Warning**

You must not change your Primary Director until all Brokers and Secondary Director have been changed.

To change to SSL connections on a Manager:

1. Copy `server.keystore` into the `DS_DATA/certs` directory of your Directors.
2. If using a self-signed certificate, copy `ssl.keystore` into the `DS_DATA/certs` directory of your Brokers and Secondary Directors.
3. Go to **Admin > System Admin > Manager Configuration > Security**, and update the SSL KeyStore Password on all Directors.
4. Perform a Manager Reinstall
5. In the **Local Configuration** step, set SSL Enabled to true.
6. Under Broker and Secondary Director Login, set SSL Enabled on Directors to true.

SSL On Large Grids

The Apache Tomcat AJP Connector can support many keep-alives (KA); however, the KA timeout is set to a default value of 60 seconds to reduce load on the Manager. This means that any time a new connection is necessary, the client must perform an SSL handshake to establish the HTTPS connections, which can impact performance.

Drivers poll on a regular basis, so they can maintain their connections. However, Engines can lose their connections when working on long tasks if the heartbeat is longer than the KA timeout. Idle Engines also lose connections on large grids where the Engine Wait time is longer than the KA timeout.

If you require HTTPS for Engines and you have a large grid with many Engines, you might notice Broker performance issues. Consider increasing this timeout to alleviate this issue. However, you must also monitor Broker performance, as this increases the number of connections that the AJP connector must poll.

Keep-alive is configured in Engine Configurations, located in the Administration Tool at **Grid Components > Engines > Engine Configurations.** The property to change is **File Server > Keep Alive Time (seconds)**.

Configuring a Reporting Database

The GridServer Manager can use a reporting database to store tables containing statistics for Services, tasks, Engines, and other records.

Reporting Database Requirements

To configure a reporting database, you must have a supported third-party database. For a list of supported databases, see the readme file included with GridServer.

Configuring the Reporting Database

To configure the reporting database, first install your Manager; second, configure the database; and third, configure the Manager to use the new database. Then, restart the Manager.

Install the Manager

1. Install and configure your GridServer Manager. Restart the Manager after completing the installation.
2. Ensure that the `tables.<db>.sql` file needed by the `createdb` script is in the `DS_MANAGER/webapps/livecluster/WEB-INF/etc/db` directory.

Configure the database

Procedure

1. Set the `CLASSPATH` on the Manager containing the Director that communicates with the database. The `CLASSPATH` must include the database driver JAR files applicable to your database. For example, to create an SQL Server database in a UNIX shell:

```
export
   CLASSPATH=JDBCDriver/msbase.jar:JDBCDriver/mssqlserver.jar:\
   JDBCDriver/msutil.jar
```

replacing *JDBCDriver* with the actual path to the directory containing your driver JAR files.

2. Run the `createdb` script provided in `DS_MANAGER/webapps/livecluster/WEB-INF/etc/db` directory on your Manager. Pass the name of the database properties file as an argument. Before you run the script, change the property file settings to match those of your database.

Configure the Manager

Procedure

1. Log in to the GridServer Administration Tool with an account that has a Security Role with the **Manager Reconfigure** feature enabled.
2. Go to **Admin > System Admin > Manager Configuration** > **Database**.
3. Configure the values in the **Reporting Database Connection** section using the appropriate table below. Note that *hostname* is the hostname and *database* is the name of the database.

Property	Value
Oracle	
URL	`jdbc:oracle:thin:@`*hostname*`:1521/`*database*
Driver	`oracle.jdbc.OracleDriver`
Transaction Isolation	`TRANSACTION_READ_COMMITTED`
Microsoft SQL Server	
URL	`jdbc:sqlserver://`*hostname*`:1433;DatabaseName=`*dbName*`;SelectMethod=cursor`

Property	Value
	(`SelectMethod=cursor` is required)
Driver	`com.microsoft.sqlserver.jdbc.SQLServerDriver`
Transaction Isolation	`TRANSACTION_SERIALIZABLE`
PostgreSQL	
URL	jdbc:postgresql://*hostname*:5432/*dbName*
Driver	org.postgresql.Driver
Transaction Isolation	`TRANSACTION_READ_COMMITTED`

4. Copy the corresponding JDBC driver’s JAR file into the Manager’s `DS_MANAGER/webapps/livecluster/WEB-INF/lib` directory.
5. Restart the GridServer Manager.

Deploying With Openshift

GridServer supports the use of the OpenShift platform for deployment and management of GridServer components in containers. The GridServer SDK provides example files to deploy Engines in OpenShift.

Introduction

GridServer supports the use of the OpenShift container platform for deployment and management of GridServer components in containers. The GridServer SDK provides example files to deploy Engines in OpenShift.

About OpenShift

OpenShift is an application platform built for running and managing containers using Kubernetes. OpenShift handles application lifecycle management functionality, such as automated building and deployment.

Deployment can be run in different models, including products for hosted public cloud, private cloud, or using on-premise infrastructure.

For more information about OpenShift, see `https://www.openshift.com/`.

About Kubernetes

Kubernetes is an open-source container. It handles the deployment, scaling, and maintenance of Docker containers. OpenShift uses Kubernetes for the container orchestration and management layer of its platform, using Red Hat Enterprise as the container host, and adding the application lifecycle management layer.

For more information about Kubernetes, see `https://kubernetes.io/`.

About the GridServer Containerization

The GridServer SDK includes an example Dockerfile which shows how to containerize a GridServer Engine to run in Docker. This enables you to deploy GridServer engines to the Openshift platform.

This example is in the SDK in the `examples/container/gridserver` directory. The `Dockerfile` is in the `engine` directory.

Deploying the GridServer Engine Dockerfile into OpenShift

The following procedure deploys an Engine to the OpenShift environment with the OpenShift CLI using the `oc` command.

Prerequisites

- This process assumes you are familiar with OpenShift. See the OpenShift web site at `https://www.openshift.com/` for more information.
- Obtain an OpenShift account and login credentials from the OpenShift web site.
- Install the OpenShift CLI. This is available from the Red Hat customer portal after you have created an account.
- Download the GridServer SDK from the GridServer Administration Tool.
- Install git on your local workstation.
- Add the Engine Dockerfile to your git repository.

  It’s also suggested to add all the folders under the `container` directory of the GridServer SDK into the git repository.

Procedure

1. Log in to OpenShift with following command:

   ```
   oc login
   ```

2. Create a new project:

   ```
   oc new-project ProjectName --description="your description"\
     --display-name="your display name"
   ```

For example:

```
oc new-project gs-engine --description="GridServer Engine \
   Container Deployment" --display-name="GridServer Engine"
```

3. Create a new application:

   ```
   oc new-app your_dockerfile_git_repository --build-env \
      GS_ARCHIVE_URL=location_of_Engine_archive_file \
      DIRECTOR_URL=location_of_GridServer_Primary_Director
      ENGINE_CONFIG=Engine_Configuration
   ```

For example:

```
   oc new-app https://git.example.com/tibco/gridserver.git \
      --context-dir=engine --build-env \
      GS_ARCHIVE_URL=http://192.168.0.100:8080/livecluster/public_
html/register/install/unixengine/DSEngineLinux64.tar.gz \
      DIRECTOR_URL=192.168.0.100:8080 ENGINE_CONFIG=default
```

Note that if the GridServer Engine is running in an OpenShift cluster with its internal IP address and is not visible by the Driver or Manager, you need to create a route in the OpenShift project, and then configure the File Server URL in the Engine Configuration.

For example: create a route in OpenShift project as:

```
http://gs-engine.gs-engine.cluster-proxy-host
```

and then set **Engine Configuration > File Server > File Server URL** to `http://gs-engine.gs-engine.cluster-proxy-host`.

This routes external communication from the GridServer Driver and Manager to the Engine internal file server (the default port is 27159).

After Deployment

After deployment, you can perform application lifecycle management tasks with the OpenShift CLI. See the OpenShift documentation for more details.

Configuring Scheduler Instrumentation

The GridServer Manager can use the `scheduler_info` table, which contains details of GridServer Scheduler operations.

Scheduler Instrumentation Requirements

To configure a database for Scheduler Instrumentation, you must have a supported third-party database. For a list of supported databases, see the readme file included with TIBCO GridServer®.

To configure Scheduler Instrumentation

To configure Scheduler Instrumentation, you must perform the following steps:

1. Install the GridServer Manager and GridServer Engine
2. Task A Configure the Database
3. Task B Configure the Manager (to use the database)
4. Restart the Manager (The JDBC JAR file is picked up by the Manager only after a restart.)

Task A Configure the Database

Create the `scheduler_info` table in any supported database (Oracle, PostgreSQL, MSSQL). Refer to the syntax provided in the `scheduler-info-readme`, available in the `DS_MANAGER/webapps/livecluster/WEB-INF/etc/db` directory.

Task B Configure the Manager

1. Log in to the GridServer Administration Tool with an account that has a Security Role with the **Manager Configuration Edit** feature enabled.
2. Go to **Admin > System Admin > Manager Configuration > Services > Scheduler Tracker**.

3. Configure the values in the **Scheduler Tracker** section on Broker by using the appropriate values given in the following table:

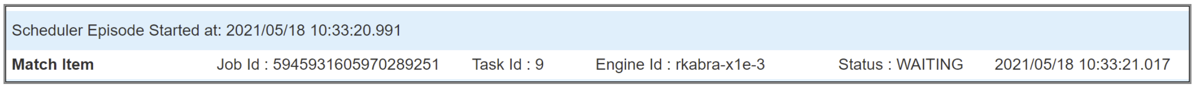
   Note that the *`hostname`* is the name of the host and *`dbName`* is the name of the database.

Property	Value
Oracle	
Database URL	jdbc:oracle:thin:@*hostname*:1521/*dbName*
Database Driver	oracle.jdbc.OracleDriver
Microsoft SQL Server	
Database URL	jdbc:sqlserver://*hostname*:1433;DatabaseName=*dbName*; SelectMethod=cursor (SelectMethod=cursor is required)
Database Driver	com.microsoft.sqlserver.jdbc.SQLServerDriver
PostgreSQL	
Database URL	jdbc:postgresql://*hostname*:5432/*dbName*
Database Driver	org.postgresql.Driver

4. Copy the corresponding JDBC driver's JAR file to the following Manager's directory: `DS_MANAGER/webapps/livecluster/WEB-INF/lib`.
5. Restart the Gridserver Manager.

To view the data, you require an account that has Security Role with the Scheduler Instrumentation feature enabled. The data can be viewed from the UI at **Diagnostics > Scheduler Instrumentation** on the Broker.

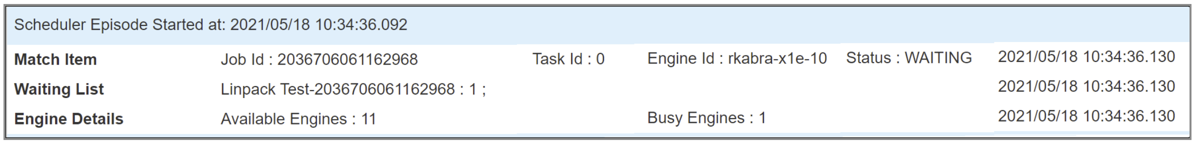
Level 1



Scheduler Episode Started at: 2021/05/18 10:33:20.991

Match Item	Job Id : 5945931605970289251	Task Id : 9	Engine Id : rkabra-x1e-3	Status : WAITING	2021/05/18 10:33:21.017

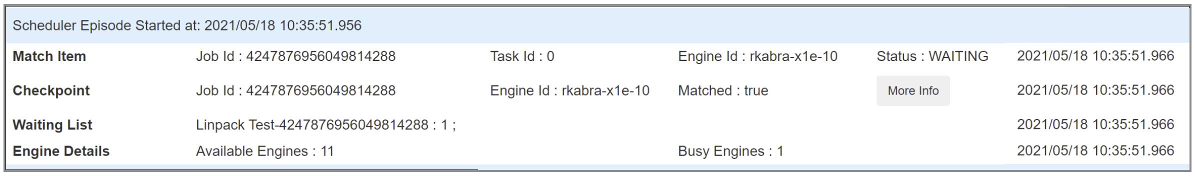
Level 2



Scheduler Episode Started at: 2021/05/18 10:34:36.092

Match Item	Job Id : 2036706061162968	Task Id : 0	Engine Id : rkabra-x1e-10	Status : WAITING	2021/05/18 10:34:36.130
Waiting List	Linpack Test-2036706061162968 : 1 ;				2021/05/18 10:34:36.130
Engine Details	Available Engines : 11		Busy Engines : 1		2021/05/18 10:34:36.130

Level 3



Scheduler Episode Started at: 2021/05/18 10:35:51.956

Match Item	Job Id : 4247876956049814288	Task Id : 0	Engine Id : rkabra-x1e-10	Status : WAITING	2021/05/18 10:35:51.966
Checkpoint	Job Id : 4247876956049814288	Engine Id : rkabra-x1e-10	Matched : true	More Info	2021/05/18 10:35:51.966
Waiting List	Linpack Test-4247876956049814288 : 1 ;				2021/05/18 10:35:51.966
Engine Details	Available Engines : 11		Busy Engines : 1		2021/05/18 10:35:51.966

Configuring Engine Instrumentation

The GridServer Manager can use the `engine_ins` table, which contains details of the Engine balancing process.

Engine Instrumentation Requirements

To configure a database for Engine Instrumentation, you must have a supported third-party database. For a list of supported databases, see the readme file included with TIBCO GridServer®.

To Configure Engine Instrumentation

To configure Engine Instrumentation, you must perform the following steps:

1. Install the GridServer Manager and the GridServer Engine
2. Task A Configure the Database
3. Task B Configure the Manager(to use the database)
4. Restart the Manager (The JDBC JAR file is picked up by the Manager only after a restart.)

Task A Configure the Database

Create the table `engine_ins` in any supported database (Oracle, PostgreSQL, MSSQL). Use the syntax provided in the `engine-ins-readme` available in the `DS_MANAGER/webapps/livecluster/WEB-INF/etc/db` directory.

Task B Configure the Manager

1. Log in to the GridServer Administration Tool with an account that has a Security Role with the Manager Configuration Edit feature enabled.
2. Go to **Admin > System Admin > Manager Configuration > Services > Engine Instrumentation**.

3. Configure the values in the **Engine Instrumentation** section on Primary Director by using the appropriate values given in the following table:

   Note that the `hostname` is the name of the host and `dbName` is the name of the database.

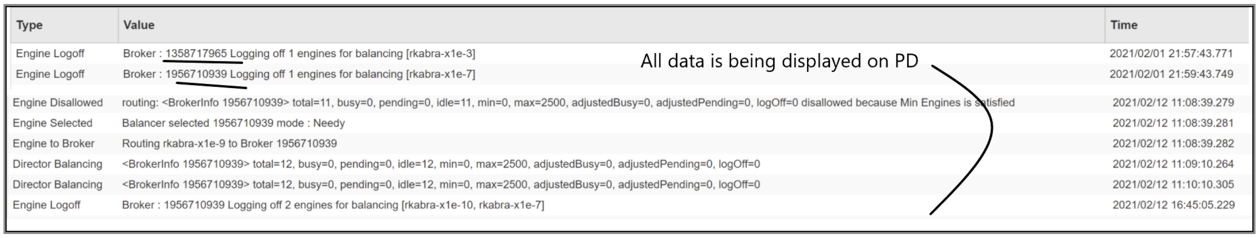
Property	Value
Oracle	
Database URL	jdbc:oracle:thin:@*hostname*:1521*dbName*
Database Driver	oracle.jdbc.OracleDriver
Microsoft SQL Server	
Database URL	jdbc:sqlserver://*hostname*:1433;DatabaseName=*dbName*; SelectMethod=cursor (SelectMethod=cursor is required)
Database Driver	com.microsoft.sqlserver.jdbc.SQLServerDriver
PostgreSQL	
Database URL	jdbc:postgresql://*hostname*:5432/*dbName*
Database Driver	org.postgresql.Driver

4. Copy the corresponding JDBC driver's JAR file to the following Manager's directory: `DS_MANAGER/webapps/livecluster/WEB-INF/lib`.
5. Restart the GridServer Manager.

To view the data, you require an account that has Security Role with the Engine Instrumentation feature enabled. The data can be viewed from the UI at **Diagnostics > Engine Instrumentation**.

On the Director

Here, you can see two Brokers: `1358717965` and `1956710939` and related information.

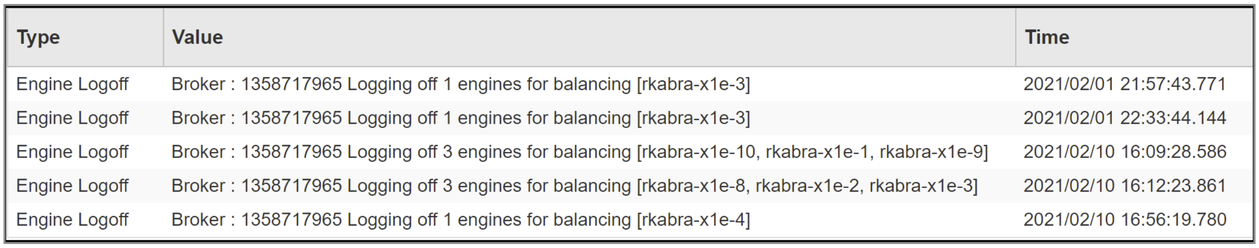


Type	Value	Time
Engine Logoff	Broker : 1358717965 Logging off 1 engines for balancing [rkabra-x1e-3]	2021/02/01 21:57:43.771
Engine Logoff	Broker : 1956710939 Logging off 1 engines for balancing [rkabra-x1e-7]	2021/02/01 21:59:43.749
Engine Disallowed	routing: <BrokerInfo 1956710939> total=11, busy=0, pending=0, idle=11, min=0, max=2500, adjustedBusy=0, adjustedPending=0, logOff=0 disallowed because Min Engines is satisfied	2021/02/12 11:08:39.279
Engine Selected	Balancer selected 1956710939 mode : Needy	2021/02/12 11:08:39.281
Engine to Broker	Routing rkabra-x1e-9 to Broker 1956710939	2021/02/12 11:08:39.282
Director Balancing	<BrokerInfo 1956710939> total=12, busy=0, pending=0, idle=12, min=0, max=2500, adjustedBusy=0, adjustedPending=0, logOff=0	2021/02/12 11:09:10.264
Director Balancing	<BrokerInfo 1956710939> total=12, busy=0, pending=0, idle=12, min=0, max=2500, adjustedBusy=0, adjustedPending=0, logOff=0	2021/02/12 11:10:10.305
Engine Logoff	Broker : 1956710939 Logging off 2 engines for balancing [rkabra-x1e-10, rkabra-x1e-7]	2021/02/12 16:45:05.229

All data is being displayed on PD

On the Broker

Here, the Broker `1358717965` contains only the data specific to it.



Type	Value	Time
Engine Logoff	Broker : 1358717965 Logging off 1 engines for balancing [rkabra-x1e-3]	2021/02/01 21:57:43.771
Engine Logoff	Broker : 1358717965 Logging off 1 engines for balancing [rkabra-x1e-3]	2021/02/01 22:33:44.144
Engine Logoff	Broker : 1358717965 Logging off 3 engines for balancing [rkabra-x1e-10, rkabra-x1e-1, rkabra-x1e-9]	2021/02/10 16:09:28.586
Engine Logoff	Broker : 1358717965 Logging off 3 engines for balancing [rkabra-x1e-8, rkabra-x1e-2, rkabra-x1e-3]	2021/02/10 16:12:23.861
Engine Logoff	Broker : 1358717965 Logging off 1 engines for balancing [rkabra-x1e-4]	2021/02/10 16:56:19.780

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

Documentation for TIBCO GridServer® is available on the TIBCO GridServer® Product Documentation page.

The following documents for this product can be found in the TIBCO Documentation site:

- TIBCO GridServer® Release Notes
- TIBCO GridServer® Installation
- TIBCO GridServer® Introducing TIBCO GridServer®
- TIBCO GridServer® Administration
- TIBCO GridServer® Developer's Guide
- TIBCO GridServer® Upgrade
- TIBCO GridServer® Security
- TIBCO GridServer® COM Integration Tutorial
- TIBCO GridServer® PDriver Tutorial
- TIBCO GridServer® Speedlink
- TIBCO GridServer® Service-Oriented Integration Tutorial

How to Contact TIBCO Support

Get an overview of TIBCO Support. You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to TIBCO Support website. If you do not have a user name, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (https://www.tibco.com/patents) for details.