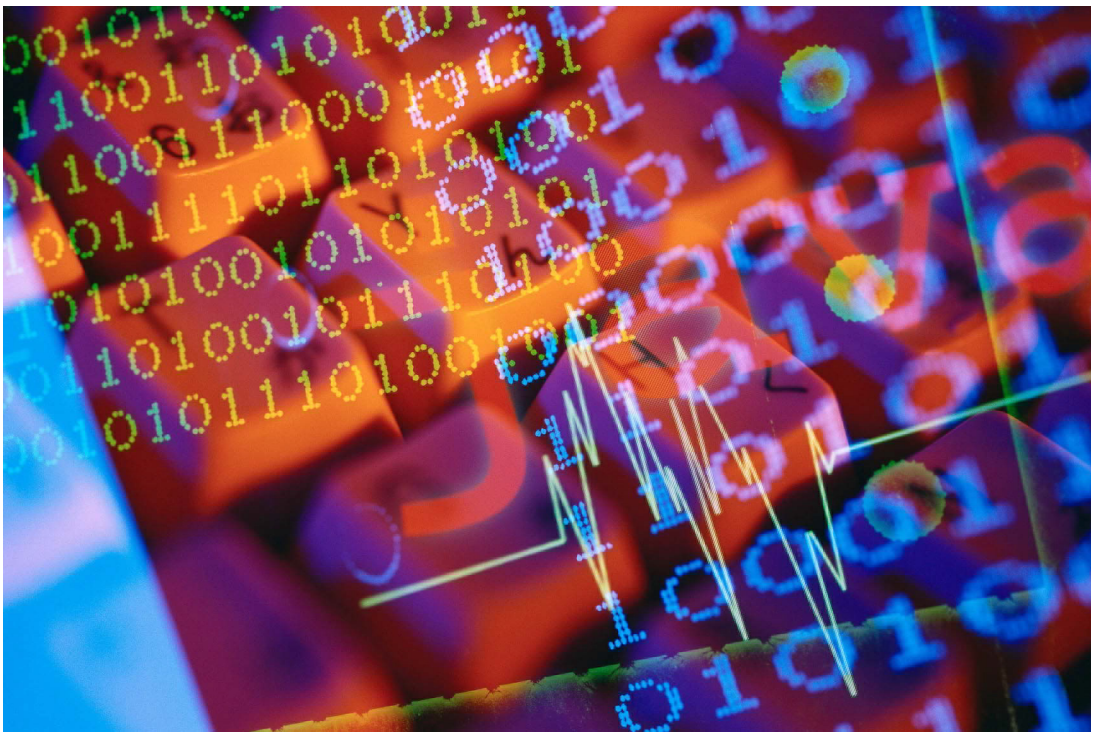
GridServer Installation Guide

Version 4.2



The GridServer Administration Series

Proprietary and Confidential

Confidentiality and Disclaimer

Neither this document nor any of its contents may be used or disclosed without the express written consent of DataSynapse. This document does not carry any right of publication or disclosure to any other party.

While the information provided herein is believed to be accurate and reliable, DataSynapse makes no representations or warranties, express or implied, as to the accuracy or completeness of such information. Only those representations and warranties contained in a definitive license agreement shall have any legal effect. In furnishing this document, DataSynapse reserves the right to amend or replace it at any time and undertakes no obligation to provide the recipient with access to any additional information. Nothing contained within this document is or should be relied upon as a promise or representation as to the future.

This product includes software developed by the Apache Software Foundation (www.apache.org/).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (www.openssl.org/).

This product includes code licensed from RSA Data Security (java.sun.com/products/jsse/LICENSE.html).

DataSynapse GridServer Installation Guide Version 4.2

DataSynapse, Inc. 632 Broadway, 5th Floor; New York, NY 10012
Tel: 212.842.8842 Fax: 212.842.8843
Email: info@datasynapse.com Web: www.datasynapse.com

For technical support issues and product updates, please visit customer.datasynapse.com.

We appreciate any comments or suggestions you may have about this manual or other DataSynapse documentation. Please send your feedback to docs@datasynapse.com.

04220102105

Contents

Chapter 1
Introduction

This guide is your starting point for planning, installing, and configuring GridServer for your site. The document is divided into several sections to explain how to install, test, and uninstall the Manager, Drivers, and Engines on both Windows and Unix platforms.

Use this guide if you are a:

- Business or technology manager planning or assessing requirements for a GridServer installation.
- Engineer or administrator installing and configuring a GridServer system.

GridServer 4.2 Documentation Roadmap

The following documentation is available for GridServer 4.2:

GridServer Guides

Four guides and four tutorials are included with GridServer in Adobe Acrobat (PDF) format. They are also available in print format. To view the guides, log in to the Administration tool, select the **Admin** tab, go to the **Documentation** page, and select a guide. A search engine is also available on this page for you to search all of the documentation for a phrase or keywords. The PDF files can also be found on the Manager at `livecluster/admin/docs`. The following guides are available:

Introducing the GridServer Platform Series:

Introducing the GridServer Platform	Contains an introduction to GridServer, including definitions of key concepts and terms, such as work, Engines, Directors, and Brokers. This should be read first if you are new to GridServer.

The GridServer Administration Series:

GridServer Administration Guide	Covers the operation of a GridServer installation as relevant to a system administrator. It includes basic theory on scheduling, fault-tolerance, failover, and other concepts, plus howto information, and performance and tuning information.
GridServer Installation Guide	Covers installation of GridServer for Windows and Unix, including Managers, Engines, and pre-installation planning.

The GridServer Developer Series:

GridServer Developer's Guide	Contains information on how to develop applications for GridServer, including information on using Services, PDriver (the Batch-oriented GridServer Client), the theory behind development with the GridServer Tasklet API and concepts needed to write and adapt applications.
GridServer Object-Oriented Integration Tutorial	Tutorial on developing applications for GridServer using the object-oriented Tasklet API in Java or C++.
GridServer Service-Oriented Integration Tutorial	Tutorial on developing applications for GridServer using Services, such as Java, .NET, native, or binary executable Services.
GridServer PDriver Tutorial	Tutorial on using PDriver, the Parametric Service Driver, to create and run Services with GridServer.
GridServer COM Tutorial	Tutorial explaining how client applications in Windows can use COMDriver, GridServer's COM API, to work with services on GridServer.

Other Documentation and Help

In addition to the GridServer guides, you can also find help and information from the following sources:

GridServer Administration Tool Help Context-sensitive help is available throughout the GridServer Administration Tool by clicking the help icon located on any page. This provides reference help, plus how-to topics.

API Reference Reference information for the GridServer API is provided in the GridServer SDK in the `docs` directory. The Java API information is in JavaDoc format, while C++ documentation is presented in HTML, and .NET API help is in HTMLHelp. You can also view and search them from the GridServer Administration Tool; log in to the Administration Tool, click the **Admin** tab, and select the **Documentation** link.

Knowledge Base A searchable archive of known issues and support articles is available online. To access the DataSynapse Knowledge Base, go to the DataSynapse customer extranet site at `customer.datasynapse.com` and log in. You can also use this site to file an issue report, download product updates and licenses, and view documentation.

Document Conventions

Convention	Explanation	Example
italics	Book titles	The *GridServer Developer's Guide* describes this API in detail.
"Text in quotation marks"	References to chapter or section titles	See "Preliminaries."
bold text	Emphasizes key terminology	Client applications (**Drivers**) submit work to a central **Manager**.
	Interface labels or options	Enter your URL in the **Address** box and click **Next**.
`Courier New`	User input, directories, file names, file contents, and program scripts	Run the script in the `/opt/datasynapse` directory.
Blue text	Hypertext link. Click to jump to the specified page or document.	See the *GridServer Developer's Guide* for details.
`[GS Manager Root]`	The directory where GridServer is installed, such as `c:\datasynapse` or `/opt/datasynapse`.	The Driver packages are located in `[GS Manager Root]/webapps/livecluster/WEB-INF/driverInstall`

Chapter 2
Installation Overview

Introduction

The GridServer installation is divided into different components. Depending on your configuration, you may need to install some or all of these components on one or more machines. Therefore, this installation guide is also divided into chapters containing installation instructions for each of these components.

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

This chapter provides a complete roadmap of what steps are required before and during a GridServer installation. It is intended to assist developers, system architects, and network architects in planning a GridServer production deployment. Read this chapter carefully before you begin installation, and determine what steps you will need to complete to meet all requirements and prepare for installation.

1) Know the Basics

This installation guide assumes a basic familiarity with the operating concepts of GridServer. The *Introducing the GridServer Platform* guide is a technical overview of the GridServer component architecture and principles of operation. This guide is included in your installation package.

You should also be familiar with Windows and Unix operating systems and TCP/IP networking.

2) Meet GridServer System Requirements

Before starting your GridServer installation, you should determine if the computers used meet the GridServer system requirements. The hardware and software needed to run GridServer are as follows:

☐ Minimum Hardware Requirements

The following hardware is required to run GridServer:

Platform	Minimum Engine/Driver	Recommended Engine/Driver	Minimum Manager	Recommended Manager
Intel Platforms	1.2 GHz PIII, 512MB RAM; 500MB+ free disk plus application disk space.		Dual 1.2 GHz PIII or 2.4 GHz P4, 1GB RAM; 5GB+ free disk space.	dual 1.2 GHz PIII or 2.4 GHz P4, 2 GB RAM recommended; 5GB+ free disk space.

Platform	Minimum Engine/Driver	Recommended Engine/Driver	Minimum Manager	Recommended Manager
Sun SPARC platforms	900 MHz UltraSPARCIII, 512MB RAM;500MB+ free disk space plus application disk space.	1.0 GHz UltraSPARCIII, 512MB RAM;500MB+ free disk space plus application disk space.	dual 1.0 GHz UltraSPARCIII, 1GB RAM;2GB+ free disk space.	dual 1.0 GHz UltraSPARCIII, 2 GB RAM recommended; 2GB+ free disk space.
zSeries Linux Platforms	512MB RAM; 500MB+ free disk plus application disk space.		768 MB RAM, 2GB+ free disk space	1 GB RAM recommended.; 2GB+ free disk space.

☐ Operating Systems Requirements

One of the following operating systems is required to run GridServer:

Engine	Manager	Driver
Windows 2000 Professional or Server, Windows XP Professional, or Windows 2003 Server.	Windows 2000 Professional or Server, Windows XP Professional, or Windows 2003 Server.	Windows 2000 Professional or Server, Windows XP Professional, or Windows 2003 Server.
Solaris 8, 9 or 10.	Solaris 8, 9 or 10.	Solaris 8, 9 or 10.
Red Hat Linux 7.1, 7.2, 7.3, 8.0, and 9.0; Red Hat Enterprise Linux 3 or 4; or SUSE SLES 8 or 9.	Red Hat Linux 7.1, 7.2, 7.3, 8.0, and 9.0; Red Hat Enterprise Linux 3 or 4; or SUSE SLES 8 or 9.	Red Hat Linux 7.1, 7.2, 7.3, 8.0, and 9.0; Red Hat Enterprise Linux 3 or 4; or SUSE SLES 8 or 9.

Note: Windows 95, Windows 98, Windows ME, Windows NT 4.0, Solaris 2.6, Solaris 7, and Red Hat Linux 7.0 are not supported; C++ compilation is not supported in Red Hat 8.0.

For RedHat 9, Red Hat Enterprise Linux 3, and SUSE SLES 8.x, when not using the 1.4.2 Engine JRE, or when running GCC 2.96 compiled code under CPPDriver on RedHat, the following pre-requisites apply:

- Red Hat 9: the `compat-libstdc++` RPM must be installed.
- SUSE 8.x: the `compat` RPM must be installed.
- Red Hat Enterprise Linux 3: the `compat-libstdc++` RPM must be installed and you must use Sun JVM 1.4.2; a repackaged Sun JVM 1.4.2 for the Engine can be obtained on request from the DataSynapse support engineering team.

Red Hat Enterprise Linux and SuSE 8/9 require the use of GCC3 versions of PDriver and CPPDriver.

Unix systems running Engines require X11 libraries to be installed.

Future GridServer releases will support operating systems versions for which the applicable vendor provides both active support and standard support. Active support means that bug fixes and security fixes are provided on a timely basis. Standard support means that these fixes are provided as part of the vendor's base support package, without requiring any additional surcharge. This minimum level of support is currently met by the following OS vendor support levels:

- Microsoft: Mainstream Support
- Sun: Vintage Phase 1 Support
- RedHat: Maintenance and Errata Support
- Novell: Maintenance Phase Support

DataSynapse may change this operating system version support policy in the future at its option, at any time, without prior notice, due to changes in OS vendor support policies or other technical or business considerations.

☐ Java Library Requirements

GridServer requires a Java 2 Standard Edition (J2SE) SDK installed on the GridServer Manager machine. Note that the Java JRE is no longer supported; you must use the SDK. Also, each Driver system requires a copy of the J2SE SDK. The Sun SDK 1.4.2_05 or SDK 1.5 (including latest patch levels) should be used for all components.

Sun's Java 2 SDK, Standard Edition (J2SE) is available at http://java.sun.com. You must first accept Sun's license agreement to download them.

Running or hosting other applications in the same Java VM as GridServer is not supported.

☐ .NET Framework Requirements

If you use the .NET Driver, GridServer requires the .NET Framework 1.1 to be installed on GridServer Engine machines. Note that the 1.0 Framework is no longer supported. Also, to use SSL with the .NET Driver, you need .NET Framework 1.1 SP1.

☐ Network Infrastructure Requirements

Your network must be configured to meet the following requirements:

Hostname resolution

GridServer uses HTTP for peer-to-peer communication between GridServer components. Each component will identify itself to other components using its IP address or its DNS name. In general, the GridServer Manager should be assigned a fixed IP address and that IP address must be resolvable through DNS. In a test environment with all Win32 systems, it's possible to use a DCHP-assigned IP address for the GridServer Manager, but this is not recommended.

Proxy configuration

GridServer does not require a proxy server, but communication between GridServer Engines, Drivers, and Managers has been verified through both firewalls and proxy servers. Communication between Brokers and Directors is via TCP, so proxies are not supported, and any intervening firewalls must have the Broker login TCP port (typically 2000) open. NAT is supported; see See Chapter 2, “Installation Overview” for more details.

Network bandwidth

If large quantities of data are to be moved through the GridServer system, then the physical network infrastructure must support it.

☐ Web browser

A web browser is required to access the GridServer Administration Tool on the Manager. Internet Explorer 5.0 or later or Mozilla Firefox is recommended; Opera, Mozilla, and Netscape Navigator are unsupported, but functional. For test deployments, a browser will be used in many cases, so ensure that all test systems have recent browsers installed. The browser must be able to communicate with the Manager machine. Javascript must be enabled; Java Applets must be enabled for some features.

☐ Perl

Solaris versions of the Manager require perl 5.0 or later. A pre-compiled perl package is available at http://www.sunfreeware.com. Perl is already installed by default on most Linux distributions. Windows Managers do not require perl.

☐ GNU Tar

Solaris systems require GNU tar to unarchive the GridServer installation and SDK.

☐ License Requirements

The GridServer Manager requires a license file that must be loaded on each running Manager (Director, , or combination) instance. Licenses are locked to one or more hostnames, and contain an expiration date. They also specify the number of Engines, Engine daemons, type of GridServer Client (Driver) used, and Manager and Engine OS allowed as per your software license. The same license should be used for each Manager in the same grid.

3) Optimize GridServer Manager Architecture

The GridServer Manager architecture can be deployed to give varying degrees of redundancy and load sharing, depending on the computing resources available. Before installation, it’s important to ascertain how GridServer will be used, estimate the transaction volume and amount of work that will be processed, and survey what hardware and networking will be used for the installation. With this information, you can plan the architecture that will best support your needs.

First, it’s important to understand the architecture of a Manager. See the *Introducing the GridServer Platform* guide for a detailed explanation of the GridServer architecture.

The GridServer Manager consists of two entities: the GridServer Director and the GridServer Broker:

A minimal configuration of GridServer would consist of a single Manager configured with a Primary Director and a single Broker. Additional Managers containing more Brokers or Directors can be added to address concerns with redundancy, volume, or other considerations.

A Manager's configuration can initially be set during Manager installation on the **Manager Type** screen. To change Manager configuration on a running Manager, click the **Admin** Tab, click **Manager Reconfigure** and follow the screens for Manager reinstallation. You must then restart your Manager.

☐ Redundancy

Given a minimal configuration of a single Director and single Broker, Engines and Drivers will log in to the Director. In this configuration, failure of the Director (such as in the case of hardware or network failure) would mean a Driver or Engine would not be able to establish a new connection.

To prevent this, redundancy can be built into the GridServer architecture. One method is to 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 will contact the Secondary Director, which will route Engines and Drivers to Brokers in the same manner as the Primary Director. The figure to the right shows an implementation with two Managers.

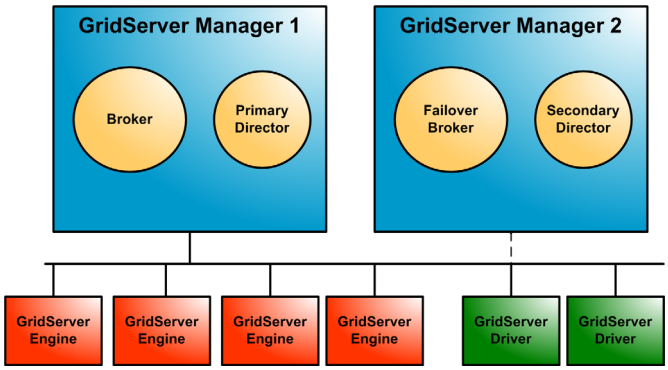


FIGURE 2-1: In this typical GridServer installation, there are a primary Director and a secondary Director, installed on two separate systems. On the same system as the primary Director, there is a live Broker. A failover Broker is installed on the same system as the secondary Director. Engines and Drivers can be added 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 also have a backup on another Manager. A Broker can be designated a Failover Broker on a Manager during installation or in the **Manager Configuration** page. If no other regular Brokers are available (because of a failure), Directors will temporarily route Drivers and Engines to Failover Brokers in the interim. The example above shows a Failover Broker on the second Manager.

For more information on redundancy, see Chapter 4, "Grid Fault-Tolerance and Failover" on page 23 in the *GridServer Administration Guide*.

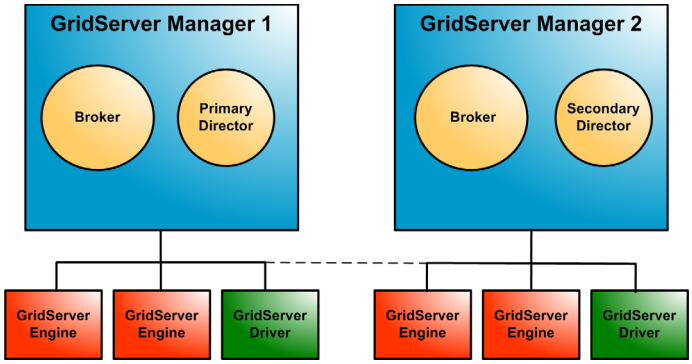


FIGURE 2-2: This GridServer installation has a Primary Director and a Secondary Director, installed on two separate systems. Live Brokers are installed on both systems. Engines and Drivers can be added to the cluster as desired. This configuration is redundant.

☐ Volume

In larger grids, the volume of Engines in the Grid may require more capability than can be offered by a single Broker. To distribute load, additional Brokers can be added to other Managers at installation. For example, the figure to the right shows a two Manager system with two Brokers. Drivers and Engines will be routed to these Brokers in round-robin fashion.

☐ Component OS and Version Interoperability

In a GridServer installation, all GridServer components must be of the same version. This is true of ManagersBrokers, Engines, and ClientsDrivers. When you upgrade a GridServer ManagerBroker, you must also upgrade ClientsDrivers.

With regard to the operating systems of components, they do not need to be running the same operating system. For example, ManagersBrokers communicating with Windows Engines need not be running Windows, and there is no performance effect, detrimental or beneficial, to be gained from running the ManagersBrokers on the same OS as any of the other components. However, it may be advantageous to run the same OS for all components, because it may make administration and troubleshooting more simple.

The one exception is that it is possible to run GridServer 4.1 Brokers with a GridServer 4.2 Director, given certain requirements are met. See “Mixed-Version Deployments” on page 54 for more details

☐ Other Considerations

Several other factors may influence how you will integrate GridServer with your computing environment. These include:

- Instead of using one Grid for all types of Services, you may wish to divide different subsets of Services (for example, by size or priority) to different Directors.
- Your network may dictate how your Manager environment should be planned. 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 could install a Manager in each location.
- Different Managers can support data used for different Service types. For example, one Manager can be used for Services accessing a SQL database, and a different Manager can be used for Services that don’t access the database.

With this flexibility, it’s possible to architect a Manager model to provide a work space that will facilitate your workload and traffic. For more information on how to best design your Manager environment, contact our Integration Services staff, and we can help you determine how to best configure your installation.

4) Configure Your Network

Since GridServer is a distributed computing application, successful deployment will depend on your network configuration. GridServer has many configuration options to help it work with your existing network. GridServer Managers should be treated the same way you treat your other mission-critical file and application servers: assign GridServer Managers static IP addresses and resolvable DNS hostnames. GridServer Engines and Drivers can be configured in several different ways. To receive the full benefit of peer-to-peer communication you will need to enable communication between Engines and Drivers (the default), but GridServer can also be configured to work with a hub and spoke architecture by disabling Direct Data Transfer.

☐ Name Service

GridServer Managers should run 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 mode is not recommended for larger deployments or heterogeneous environments. GridServer does not support the use of DHCP-assigned client addresses for Manager components; DHCP-assigned addresses may be used for Engines and Drivers.

☐ Protocols and Port Numbers

GridServer uses the Internet Protocol (IP). All Engine-Manager, Driver-Manager, and Engine-Driver communication is via the HTTP protocol. Manager components, Engines, and Drivers can be configured to use port 80 or any other available TCP port that is convenient for your network configuration. The default port is 8000.

☐ Director-Broker Communication

All Director-Broker communication is via TCP. The following ports are used, and should be open between Directors and Brokers.

HTTP communication occurs on port 8000 by default. This can be changed to another TCP port at Manager installation or with the **Resin Setup** option on the **Manager Reconfigure** page.

The default Director TCP port used by Brokers for login is 2000, but another port can be specified at installation time or with the **Manager Reconfigure** page.

After the Broker logs in, another pair of ephemeral ports is assigned for further communication. The Broker and Director can also be configured to use static ports.

☐ Manager-Engine and Driver-Manager Communication

All communication between Engines and Managers (Directors and Brokers) and between Drivers and Managers is via the HTTP protocol, with the Engine or Driver acting as HTTP client and the Manager acting as HTTP server.

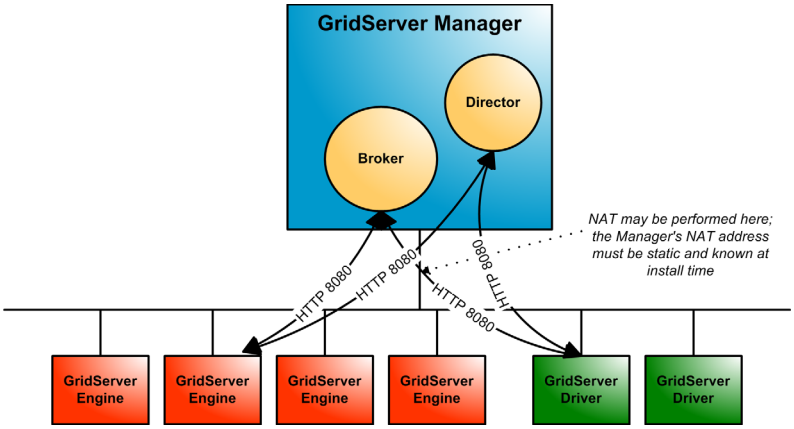


FIGURE 2-3: All communication between Engines and Managers or Drivers and Managers is via the HTTP protocol on the port assigned when the Manager is installed. In this diagram we assume that port 8080 has been chosen. All HTTP communication is initiated by the Engine or Driver as the HTTP client.

☐ Using NAT

The Manager can be configured to work with a NAT device between the Manager and the Engines or Drivers. If there are no Engines on the untranslated site of the network, you can do this 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 manual Windows, Linux, and Solaris Engine installation.

If there are clients (Engines and Drivers) on both sides of a NAT device, NAT Translation can be configured on the Manager. This enables the URLs used for Manager to client communication. To configure NAT Translation in the GridServer Administration Tool, click the **Manager** tab, click **Manager Configuration**, go to the **Communication** section, and 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. In the **NAT URL** field, type the external URL to be used 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 or Outside NAT Range** to **Inside**; to use NAT translation for clients outside the NAT Range, set this to **Outside**.

☐ Broker-Director Communication

Communication between Brokers and Directors is via TCP. NAT is not supported between Brokers and Directors. By default, the Broker will log in on port 2000, and ephemeral ports will then be assigned for further communication. This configuration does not permit a firewall or screening router between the Brokers and Directors. If a firewall or screening router must be supported between Brokers and Directors, then the firewall or screening must have the Broker login port (default 2000) open. Additionally, the Brokers must be configured to use static ports for post-login communication, and those ports must be open on the firewall as well. This option is not available by default; contact our Integration Services staff, and we can help you configure your system.

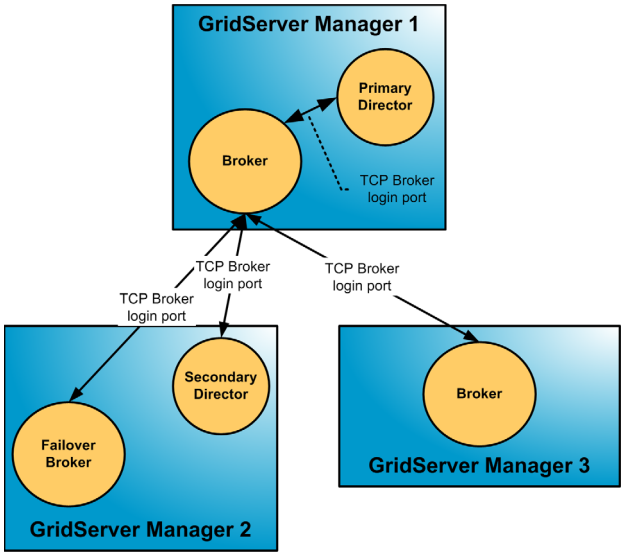


FIGURE 2-4: All communication between Brokers and Directors is via TCP on the Broker login port (default TCP port 2000)

☐ 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 inputs and outputs must be sent through the Manager. Sending the inputs and outputs through the Manager will result in higher memory and disk use on the Manager, and lower throughput overall.

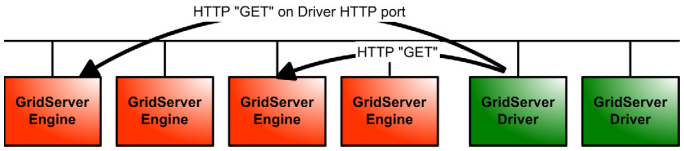


FIGURE 2-5: When Direct Data Transfer is used to move TaskInput data from Drivers to Engines, the Engines "pull" the data via HTTP directly from the Driver. The Driver's embedded HTTP server can be configured to use a static TCP port or an ephemeral TCP port according to the `driver.properties` file.

With Direct Data Transfer, only lightweight messages are sent though the Manager, and the "heavy lifting" is done by the Driver and Engine nodes themselves. Direct Data Transfer requires that each peer knows the IP address that he 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.

An optional Engine Hook can be installed to enable communication between Drivers and Engines when NAT is in use. Contact your DataSynapse Services specialist for more information on installation of this hook.

For GridServer deployments where NAT is already in effect and the optional Engine Hook is not installed, NAT between Drivers and Engines can be supported by disabling peer-to-peer communication as follows:

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 Direct Data Transfer must be disabled in the Engine configuration.

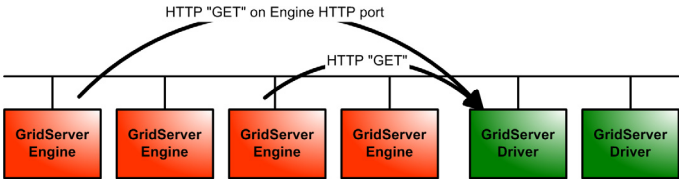


FIGURE 2-6: When Direct Data Transfer is used to move TaskOutput data from Engines to Drivers, the Drivers "pull" the data via HTTP directly from the Engines.

Likewise, if, from the perspective of the Engines, the Drivers are behind an NAT device, then the Drivers cannot provide peer-to-peer communication. If the Drivers' addresses are known in advance, they can be provided. Otherwise, Direct Data Transfer must be disabled in the Driver properties file.

☐ SSL

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

For more information on using SSL, see Chapter 9, "Configuring Security" on page 69 of the *GridServer Administration Guide*.

5) Install the GridServer Components

After you have prepared for your installation, you can install the GridServer components, as described below.

☐ GridServer Manager

To install Managers, follow the installation procedure outlined in the Manager Installation chapter. If you have multiple Managers, you will need to repeat the procedure for each Manager. You must have a running Manager before you proceed to Engine installation.

For more information on installing Managers, see Chapter 3, "Manager Installation" on page 23.

☐ GridServer Engine

GridServer Engines come in two variants: Windows Engines and Unix Engines. As these platforms are typically deployed differently, the corresponding GridServer Engine deployment strategies have different characteristics.

Example 2.1: Installation Strategies

	"One Click" Web Installation	Network Installation	SMS Installation	Manual Installation
Windows Engine	✓	✓	✓	✓
Unix Engines		✓		✓

- **One-Click Web Install** uses an ActiveX control or Java applet for ease of use when installing small numbers of Windows Engines.
- When a Manager directory can be mapped using Windows file sharing or Samba, use **Network Installation from Server** to run a `.bat` file when the Engine's OS is booted. Also, an installation script can be used with Unix distributions to install Engines remotely.
- For both Windows and Unix, **Manual Installation** is supported from `.zip` and `.tgz` files.

Security Considerations

On the Windows platform, both Engines and Managers run as services owned by the Local System account. You must reboot the machine after install for the Engine to be run by the Local System account. The Engine will then run as an automatically started service after each system reboot.

To prevent ordinary users from accessing files in the Engine's directory tree, the Windows system administrator can set up the Engine directory so that it does not inherit permission from the parent directory and it grants full access to the built-in `SYSTEM` and `SERVICE` users but does not grant access to any other users.

On Unix platforms, Engines and Managers will be installed with the file permissions of the user that installs them. Engines and Manager should be started by that same user. Typically, a Manager port above 1024 will be chosen as the Manager port for this reason.

For more information on installing Engines, see Chapter 4, "Windows Engine Installation" on page 31, and Chapter 5, "Unix Engine Installation" on page 37.

☐ GridServer Driver

The GridServer Driver is used by application programs to communicate with the GridServer Manager and thereby make use of the Grid's compute resources. For that reason, Driver deployment is usually bundled with your application deployment.

The GridServer SDK is a platform-specific archive which contains all Drivers for a specific operating system. It is typically downloaded to the development systems using a Web browser.

☐ Database

GridServer uses databases for internal use and reporting. You can configure GridServer to use an external reporting database after Manager installation. This can be a standalone HSQLDB database, which is included, or an enterprise database system.

WARNING By default, no reporting database is configured. The included HSQLDB reporting database can be used for lightweight reporting, but it is primarily for demonstration, development, and integration purposes. For production environments, use an enterprise database system that supports JDBC.

For more information on database configuration, see Appendix B, “Database Configuration” on page 59.

56) Plan Application Delivery

GridServer supports many different options for delivering your software, configuration, and data files to Engines. You can synchronize files between the Manager and the Engines by using **Grid Libraries**, or the GridServer file replication service. Or you can configure the Engines to locate key program files, such as `*.dll`, `*.so`, and `*.jar`, anywhere on the Engine’s file system.

For more information on resource deployment, see Chapter 7, “Application Resource Deployment” on page 43 of the *GridServer Administration Guide*.

Chapter 3
Manager Installation

Introduction

To begin your GridServer installation, you must install one or more Managers. This chapter covers installation of the Manager, plus installation of other Manager components such as Primary and Secondary Directors, and Primary, Secondary, and Failover Brokers.

Before installation, read the previous chapter, “Installation Overview” on page 11 thoroughly, and decide what Manager components you will install on each Manager.

About Installation

The Installation includes an embedded Resin application server, plus the GridServer software, packaged as a `tar.gz` file, and including the Resin JAR files, Resin bin files, and startup scripts. The same archive is used for Windows, Linux, and Solaris installations. Your software will be provided on a CD-ROM or downloaded install package. Make sure the package is available on your machine before you begin.

Other supported platforms are optionally available from DataSynapse customer support as a GridServer patch. These will typically include a Readme file with further instructions or requirements for installation.

WebSphere Installation

GridServer can be deployed as a WebSphere application. GridServer version 3.1.1 has been certified for WebSphere Advanced Server Edition 5.0. Other versions require confirmation. Contact a DataSynapse representative for more information.

Installing GridServer

NOTE: If you are upgrading from a previous version of GridServer, please see Chapter 9, “Upgrading GridServer” on page 49 and complete all steps required before beginning this installation process.

1) Copy files before installation

To copy files used to install GridServer, do the following:

1. Decide where to install the Manager. The recommended location for Unix machines is `/opt/datasynapse`; the recommended location for Windows is `c:\datasynapse`. This directory is referred to as `[GS Manager Root]` below. Note that you should not install the Manager in a directory 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.
2. The software is shipped as a `gzip`ed `tar` archive. To unpack it on a Windows system, use WinZip or a similar tool. On a Unix system, use tar and gzip. (Note that with Solaris, you must use GNU tar.) For Unix, use the following command:

```
gzip -d -c GridServer*gz | tar xvf -
```

NOTE: The tar file contains the `datasynapse` directory, and should be extracted in the parent directory of its destination. For example, if you expand the archive in `/opt`, then `/opt/datasynapse` will be created.

3. Set the `JAVA_HOME` environment variable to your Java root directory. (For example, `C:\j2sdk1.4.2_08`.)
4. Make sure that the `$JAVA_HOME/jre/lib/ext` directory is empty except for the four files included with the JRE (`dnsns.jar`, `ldapsec.jar`, `localedata.jar`, and `sunjce_provider.jar`) to avoid classloading issues.
5. Install the unlimited strength JCE for your Java SDK, if you have not done so previously. The files are located in `[GS Manager Root]/webapps/livecluster/WEB-INF/etc/jce`. Follow the instructions in the `README.txt` for your SDK to install the files.

2) Start the Manager

Start the Manager, as follows:

Windows

The Windows Manager can be started by running the `server.bat` file, located in the root directory of the GridServer distribution. For testing purposes, you can double-click `server.bat` to launch the Manager and subsequently start GridServer. This opens a console window containing log messages, and a control panel, which you can use to start, stop or exit the Manager.

It is also possible to run the Manager noninteractively as a Windows Service. To do so, open a command window and run `server.bat` with the following arguments:

Argument	Description
`[none],run`	Start the Manager
`install` *`name`*	Install the Manager as a service, with the name "DataSynapse GridServer Manager *name*", where *name* is optional. If *name* contains whitespace, you must enclose it in quotes.
`remove` *`name`*	Remove the Manager's service named *`name`*, where *`name`* is optional if you installed without one.

Unix

Launch the Manager by invoking the `server.sh` script in the install directory with the `start` command:

```
./server.sh start
```

Subsequently, to stop the Manager, invoke the same script with the `stop` command:

```
./server.sh stop
```

You can also run several Managers using one shared Manager directory. See “Running Multiple Managers From a Single Installation” on page 28 for more instructions.

Other optional arguments that prefix `start/stop` for the Unix Manager include the following:

Argument	Description
`-t`	Ignore and do not create `touchfile.`*`host`* files. Normally, these files are created in the Manager directory to prevent running multiple Manager instances from the same directory on the same host.
`-p`	Specifies the name and location of the `http.pid` file used by the Resin application server.

Typically, the system should be configured to launch the Manager automatically at system startup and stop it 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 on initialization and termination scripts, please refer to the `init` and `init.d` man pages on your Unix system and see “A Sample Unix rc.d Script” on page 29.

By default, the `server.sh` script for Unix systems forces the `PATH` to be `/bin:/usr/bin:/usr/local/bin`. This can pose an issue if, for instance, the Manager system has perl installed in `/opt/perl`. To retain the `PATH` setting that exists in the shell before the Manager is started, set the `DS_USE_SYSTEM_PATH` to true.

3) Initialize Your Manager

After the Manager is running, you need to initialize it. To do this one-time process, you complete the configuration in the Administration Tool, restart the Manager, and create a login account.

After starting your Manager, do the following:

1. Start the GridServer Administration Tool

   Go to `http://<yourhost>:<port>/livecluster` to open the installation page.

2. Select Installation Type

   There are two different options for installing a Manager, **Typical** and **Custom**. The default is a **Typical** installation, which will install a primary Director and a Broker with all default settings. If you select **Typical**, click **Next** and skip to step 5 below.

   If you wish to install a secondary Director, a failover Broker, or no Broker, configure Resin, or change any other parts of your configuration, choose **Custom**. You will be presented with additional configuration steps, where you can choose whether to install a new Manager, upgrade an old installation, or configure the Resin application server .

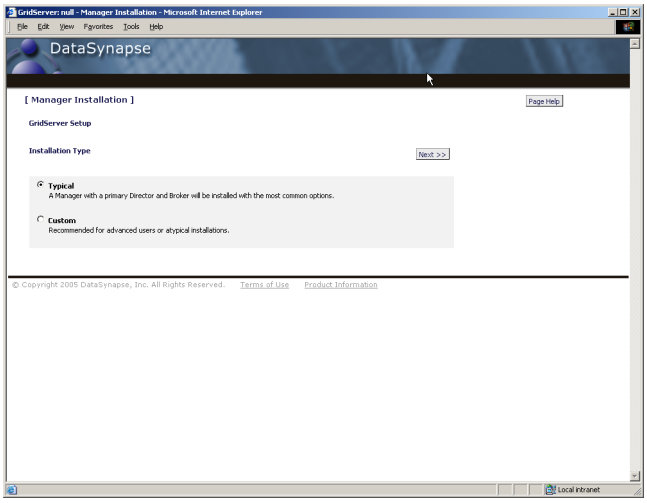


FIGURE 3-1: Selecting Installation Type.

3. Configure Resin

   If you need to make any changes to the Resin configuration (for example, if you wish to use an HTTP port other than the default of 8000, run multiple Managers on one host using unique HTTP ports, set up SSL, or run with a non-default base directory, such as for running multiple Managers from a single share) you must first configure Resin. You **must** do this before you configure GridServer. You also must choose the **Custom** setup in step 2 above to select this option.

   To configure Resin, select **Resin Configuration** from the Manager Configuration page and click **Next**. Enter values for each of the seven screens.

4. Enter Configuration Values

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

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

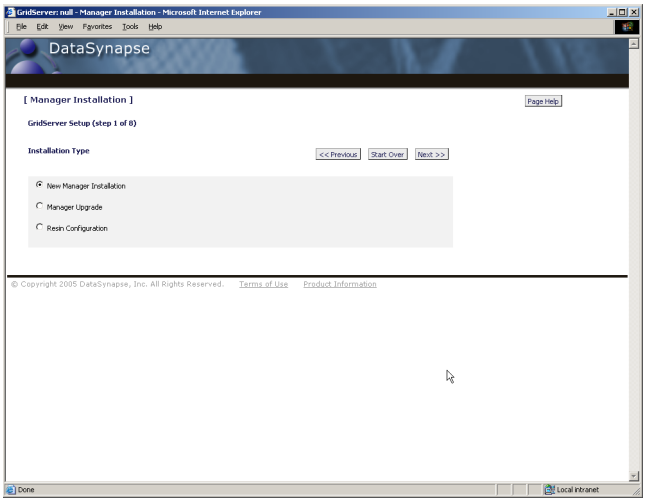


FIGURE 3-2: Selecting installation, upgrade, or Resin configuration.

Upgrades from previous GridServer Versions

NOTE: If you are upgrading from a previous version of GridServer, please see Chapter 9, “Upgrading GridServer” on page 49 and complete all steps required before beginning this installation process.

Before installation, move your previous version of GridServer to another location. On the GridServer Manager Setup page, type the path to your old installation in the **Previous Base Directory** box. Your old settings will be migrated to your new installation.

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. Note that if you install a secondary Director, you should reconfigure the Manager containing the primary Director by going to the **Admin** menu, clicking **Manager Reconfigure**, and entering the secondary Director’s address and port in the corresponding page. This will configure the primary Director to recognize the secondary Director, as well as reconfiguring Engine and Driver configurations accordingly.

5. Verify and Install

   When you have completed entering information in the **Custom** installation, or after you have selected the **Typical** information, the **Verify Setup Parameters** screen will display the parameters that will be used to install GridServer. If you believe any of these parameters are not correct, or you need to change any of the parameters, click the **Start Over** button to restart the installation process.

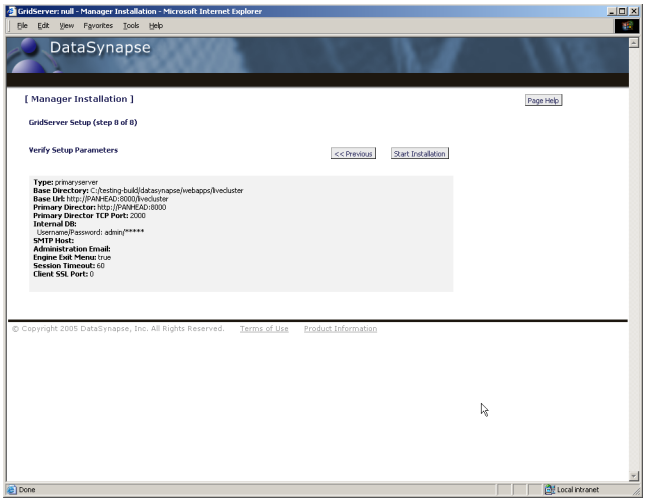


FIGURE 3-3: Verify Setup Parameters screen.

   NOTE: When installing the GridServer Manager on a Unix system, the installer may not display the fully qualified hostname as the Manager name. It may determine that the machine hostname is the full name, which may cause routing problems after installation. If this is the case, the hostname of the machine can be changed with the Custom install option. Go back to beginning of the installation screens and repeat the installation from Step 2, above, entering the correct DNS-resolvable hostname in the **Verify Hostname** screen.

   If you are satisfied with the parameter values, click **Start Installation** to begin installation.

6. Restart the Manager

   After Manager installation has completed, you will be prompted to shut down the application server. On Windows machines, click **Quit** in the control panel or simply close the console window; in Unix, use the `server.sh stop` command. Restart the Manager, using the same command you used earlier.

7. Create an account

   After Resin has completely restarted, go to `http://<yourhost>:<port>/livecluster` again. You will be prompted to enter a username, password, and password verification for your initial admin account. You can use this account to log in to the Administration Tool and create additional user accounts.

   After your account is created, you will be brought to the home page in the Administration Tool.

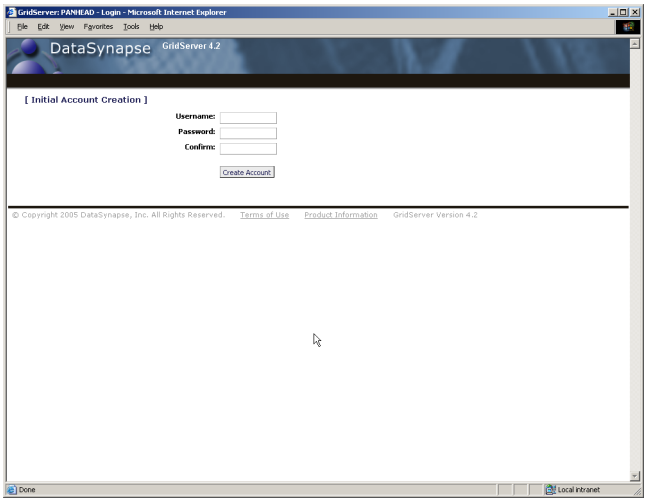


FIGURE 3-4: Initial account creation.

Running Managers From a Different Base Directory

By default, all files used by GridServer are stored under the `livecluster` directory in your applications directory. However, you may want to store all volatile files specific to an installation in a different location. The GridServer base directory refers to the location of these files.

To use an alternate location with the Resin application server:

1. Decide upon a directory that will be the new base directory location for your Manager. When specifying a new alternate base directory, the base directory should not be a child directory of the `[GS Manager Root]` installation directory. Also, the install and alternate base directory cannot be on two different Windows drives (for example, `C:` and `T:`.)
2. Make a copy of the `[GS Manager Root]/conf/resin.conf` file, and place it in the new directory location. Note that the `resin.conf` file must be writable.
3. Set the `DS_BASEDIR` environment variable to the desired base directory location, and the `RESIN_CONF` environment variable to the path and filename for the copy of the `resin.conf` file. You will need to set these each time you start or stop a Manager, so you may wish to create a small shell script to do this.
4. Start the Manager. When you connect to the GridServer Administration Tool, you will have a Resin Configuration page instead of the normal first-time installation page. Follow each of the steps for Resin configuration.
5. Re-start the Manager. You will then encounter the first-time installation page. Perform the installation.
6. From then on, set the environment variables whenever you run this Manager. Also, if you change the Engine JRE as described in "Updating the Windows Engine JRE" on page 90 of the *GridServer Administration Guide*, you will need to copy it into the different base directories, not the original directory.

Running Multiple Managers From a Single Installation

You can run more than one Manager on multiple machines from the same shared GridServer installation. To do this, you install GridServer in one shared location, but then have each Manager run from its own unique base directory on another unique machine.

To run multiple managers from a single location, use the procedure in "Running Managers From a Different Base Directory" for each manager. Start each Manager with `server.sh` or `server.bat`, and add the argument `-p pidfile`, where `pidfile` is the path and name of an arbitrary file unique to each Manager. For example, you could start the first Manager with `server.sh -p server1 start`, the second one with `server.sh -p server2 start`, and so on.

When you need to stop a manager, use the `-p pidfile` argument.

Running multiple Managers on a single machine is possible, but not recommended, due to performance issues. If you are configuring multiple Managers on a single machine, you will need a unique HTTP port (default 8000), HTTPS port if you're using SSL, and if using multiple Directors, JDBC listen ports (defaults 9157 and 9158) and Director port (default 2000) for each Manager.

Installing Into a Read-only Mount

You can also run one or more Managers from a GridServer installation in a read-only directory, with volatile files stored in another location. To do this, use the instructions "Running Managers From a Different Base Directory" above. For Unix Managers, start the Manager with `server.sh`, and add the argument `-p altdir/pidfile`, where `altdir` is the name of a directory where the Manager has read/write permission, and

pidfile is the name of an arbitrary file unique to the Manager. For example, you could start the Manager with `server.sh -p /nfs/dir/server1`, where `/nfs/dir` is a mount directory, and `server1` is an arbitrary file name.

Installing Manager Licenses

Each GridServer Manager requires a valid license to function. Licenses are limited by date, hostname, and number of Engines. By default, a demo license for four Engines is included with each Manager, but for further evaluation or production use, you must obtain a license by contacting DataSynapse Support.

You can view your Manager's license information in the GridServer Administration Tool, by clicking the **Admin** tab, then clicking the **License Information** page.

A Manager license consists of a single XML file, and is provided by DataSynapse Support. You can inspect the license with a text editor to determine its capacity, but editing the file will corrupt it.

To install the license:

1. In the GridServer Administration tool, click the **Admin** tab, then click the **License Information** page.
2. Copy the license file to a location accessible with your web browser (either a local directory or a shared directory.)
3. Click **Browse**.
4. Find the license file and click **Open**.
5. Click **Upload New License**.

If the license file is valid, it will overwrite the existing license and changes will take place immediately. If it is expired, corrupt, or otherwise not valid, an error message will appear and your existing license will remain in place.

A Sample Unix rc.d Script

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

Example 3.1: `dsserve` script

```
#!/bin/sh
# Startup script for DataSynapse Manager
#

# Source function library.
. /etc/rc.d/init.d/functions
```

Example 3.1: `dsserve` script (Continued)

```
prog="server"
DSBASE=/opt/datasynapse
JAVA_HOME=/usr/local/java
export JAVA_HOME

case "$1" in
       start)
           cd $DSBASE
           ./server.sh start
           ;;

       stop)
           cd $DSBASE
           ./server.sh stop
           ;;

       restart)
           cd $DSBASE
           ./server.sh stop
           ./server.sh start
           ;;

       *)
           echo $"Usage: $0 {start|stop|restart|condrestart|status}"
           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 is run and passed 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 dictate the order in which scripts run.

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

Example 3.2: links created in rc*.d directories

```
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
```

Chapter 4
Windows Engine Installation

GridServer Engines for Windows are installed by logging in to the GridServer Administration Tool and clicking a link for 1-Click Install. This downloads and starts an InstallShield Wizard to install and configure the Engine on a system.

In some situations, the 1-Click install may be impractical. A manual installation enables you to download an Engine and install it without the wizard, or transfer it to another machine before installation. You can also use a network installation to install Engines on many machines, or install the Engine on a Windows Terminal Server for use by multiple PCs.

1-Click Install

NOTE: 1-Click install will not work if you are accessing the Manager using SSL (through an HTTPS URL.)

To install Engines on Windows systems:

1. Ensure the Manager has been installed correctly and is up and running.
2. Go to the machine on which you wish to install the Engine, and log in to the GridServer Administration Tool.
3. Click the **Engine** tab, then click **Engine Install** to display the screen to the right.

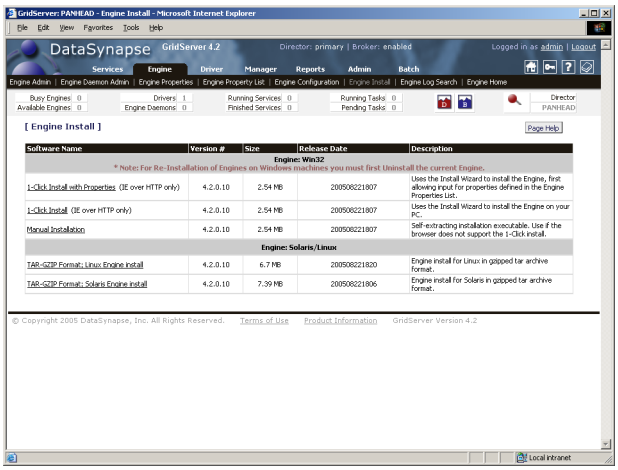
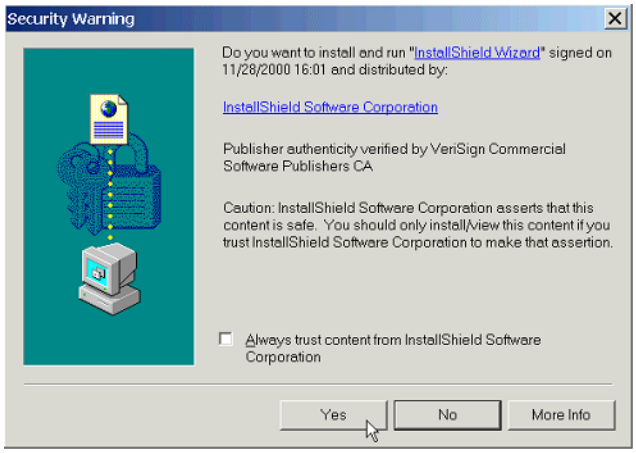


FIGURE 4-1: The **Engine Install** page.

4. Click **1-Click Install**.

   The InstallShield Wizard will then start, and prompt you to verify the authenticity of the Installer. Click **Yes** to continue.



5. When your authenticity is verified, click **Next**.

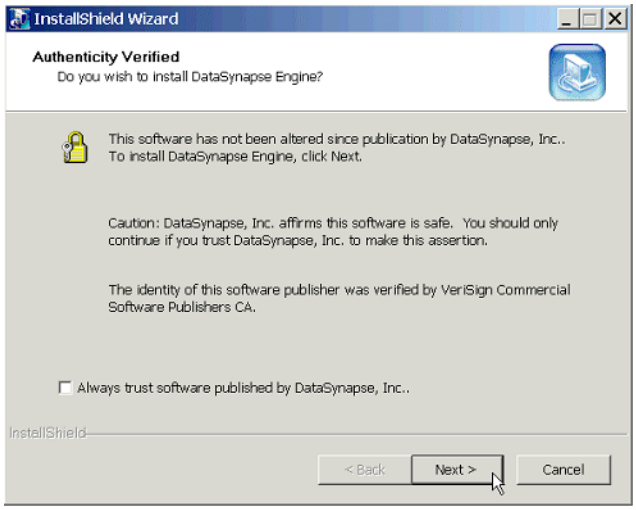


FIGURE 4-2: Verifying authenticity of the installer.

6. Choose the installation folder to hold the Engine executable and supporting files; the default location is: `C:\Program Files\DataSynapse\Engine`.

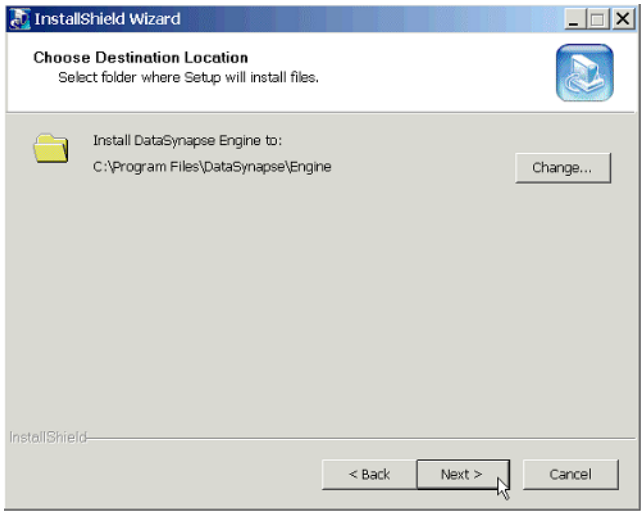


FIGURE 4-3: Choosing a destination location for the Engine.

When the wizard is complete, a red DataSynapse icon displays in the TaskTray (usually located at the bottom right corner of the taskbar).

FIGURE 4-4: The red Engine icon, showing a stopped Engine.

7. After the icon displays, the Engine spends a minute or two downloading additional files from the Manager. When downloading completes, the Engine is ready to take work the next time your computer becomes idle. To preemptively engage the Engine and start work when your computer is not idle, right-click the icon and click **Start**.

   If you engage the Engine, the icon should display green, then yellow, indicating an idle Engine. You can stop the active Engine right-clicking the icon and clicking **Stop** to manually deactivate the Engine.

   Once installed, the Engine activates and logs in to the Manager whenever it detects an idle condition on its host processor.

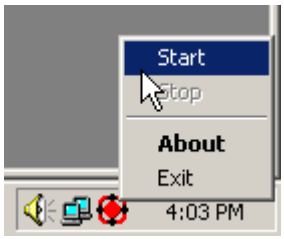


FIGURE 4-5: Starting an Engine.

1-Click Install with Properties

Engines can also be installed with a set of properties defined, which can then be used for discrimination.

This is identical to 1-Click Install as described above, except at step 4, click **1-Click Install with Properties**.

If you click **1-Click Install with Properties**, you must first enter values for each of the parameters defined in the **Engine Property List** page. The default Engine property list is shown to the right, and includes User Name, Location, Group, and Description.

Note that Engine Property values cannot contain commas.

Property	Value	Description
User Name		hostname of the machine where the engine is being installed
Location		machine location
Group		work group to attach engine
Description		brief description of machine
		Submit

FIGURE 4-6: The default Engine property list.

Manual Installation

You can also manually install an Engine. To do this, click the **Manual Installation** link instead of a 1-Click install link from the **Engine Install** page. This will download a file named `DSIntranetInstall.exe`, which you can either run or save and move to another machine.

When you run `DSIntranetInstall.exe`, you will be asked to enter a Manager name. Enter the URL of your Primary Director, such as `http://hostname.example.com:8000`.

Network Installation

Engines can also be installed onto multiple PCs from a shared drive via a login or update script.

To set up a network installation, do the following:

1. Copy the contents of `[GS Manager Root]/webapps/livecluster/WEB-INF/engineInstall/win32/network` to a shared directory, such as `/share/datasynapse`.
2. In a login or update script include the following:

```
cd /share/datasynapse
install.bat
```

You can also modify the `install.bat` file to install Engines in either the system or a user account. `install.bat` runs the program `setup.exe`, which can take username arguments.

The `setup.exe` program takes a username and password argument before other installation arguments. The username argument in the form of `DomainName\UserName`, or can be `.\UserName` if the account belongs to the built-in domain.

For example, the following command will silently install the Engine and run the Engine Daemon service with the account `.\dave` with the password `joshua7`, writing a log file to `c:\datasynapse.log`:

```
Setup.exe .\dave joshua7 -s -f2c:\datasynapse.log
```

Note that when running the Windows Engine Daemon service as a user other than Local System, the user account running the DataSynapse Engine service will be granted the Logon As a Service (`SeTcbLogonService`) right by the installer, provided the domain name given does not end with ".". The user account must also have read access to perflib in the registry as mentioned below in "Configuring Permissions for Processor Utilization Mode" on page 34.Also, UI Idle mode doesn't work unless DSHook is installed, which is detailed in Appendix A, "Using DSHook" on page 57.

Installing Windows Engines in a Non-default Location

By default, the Windows Engine installer will install Engines at `C:\Program Files\DataSynapse\Engine`. To install an Engine manually or from a network script in another location, do the following.

1. Go to the location of the Windows Engine network installer, which is `[GS Manager Root]/webapps/livecluster/WEB-INF/engineInstall/win32/network`.
2. Edit the `setup.iss` file.
3. Change the value of the `szDir` property to another location.
4. Copy the contents of the directory to the install destination and run `install.bat`.

Engine Permissions

When installation of the Engine is done by an administrator, but the Engine runs as a non-admin user thereafter, there may be permissioning issues, especially if the Engine removes a key and creates it on an update event. This section contains a list of keys created and used by Engines, which will aid you in determining what permissions should be set for Engines to function properly.

The following 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 will not work correctly, because the Engine Daemon will not be able to retrieve performance data (such as CPU information) from the OS.

To remedy this, do the following:

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 is logged in as, and grant Read access.

This process must be done for each Engine in the Grid.

NOTE: Novell has a free utility called `regperm.exe` that can let you script permission-setting in the registry that works for remote machines as well. It can be found at `http://www.novell.com/coolsolutions/tools/1129.html`. The following command line works for changing the key:

```
regperm /k "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib" /e
/a:datasynapse\test:r /i
```

Configuring Permissions for User Idle Mode

If you install an Engine Daemon to run as a Service and the Service logs in as another user, the User Idle Mode will not work correctly, because the Engine will not be able to retrieve UI information from the system.

To properly use User Idle mode, you must run DSHook, a small service that runs in the Local System account and passes information to your Engine Service.

To install DSHook, see Appendix A, “Using DSHook” on page 57.

Configuring Engines

To change your Engine’s settings (for example, to point them to a new Director), you change the profile, or configuration used by the Engine. All Engines using that configuration will also change configuration. To change an Engine configuration, go to the **Engine Configuration** page of the GridServer Administration Tool, select an Engine configuration, and change its values**.**

WARNING When adding values to the **Environment Variables** box, it is possible to set variables, particularly `PATH`, that can cause an Engine to fail to start.

When new Engines are installed, they are assigned the default configuration for their platform (Windows, Linux, or Solaris) and will have the configuration defined for that profile.

You may configure groups of Engines with different properties by creating a new configuration on the **Engine Configuration** page of the GridServer Administration Tool, and assigning it to Engine from the **Engine Daemon Admin** page. You could, for example, create a second configuration for a group of Windows machines that points to a different Director and assign it to a subset of your existing Windows Engines.

Configuring Dual-Homed Engines

In some network configurations, the PC running an Engine may have dual network interfaces, and the Engine may default to using the incorrect interface for its fileserver. To configure the Engine to use a different network interface, select the Engine Configuration that will be used by the Engine on the **Engine Configuration** page, and set the **Net Mask** value under the **File Server** heading to match the network range on which the Engine should run.

Running Engines in Console Mode

Windows Engines can be run from console mode, which enables you to see logs interactively as the Engine is running.

To run an Engine from console mode:

1. Open a commandline prompt by clicking **Start**, selecting **Run**, then entering `cmd`.
2. Change to the directory containing your Engine installation, usually `C:\Program Files\DataSynapse\Engine`.
3. Type `engine.exe -console` to run the Engine in console mode.

The Engine will run, with log information displayed in a new console window.

Chapter 5
Unix Engine Installation

To install Engines on Unix systems, you must download and install an Engine on each system, and run a configure script. The script will create a profile on the Manager that can be used for other Engines, or you can use a profile on the Manager that has already been created.

Installing the Engine

The GridServer Unix Engine is packaged as a `gzip`-compressed `tar` archive. The Engine installation process installs the files on the Unix system, creates or uses a profile on the Manager, and associates the Engine with that profile.

To install an Engine:

1. Download the installation to a Unix machine from the **Engine Install** page in the GridServer Administration Tool.
2. 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 -
```

3. Run the `configure.sh` script located in the directory in which you unpacked the archive. You should only run the script once, and not once per Engine.

The `configure.sh` arguments are summarized below:

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 host-specific directories for native code logging and configuration info to be created. The default value is `./profiles`
`-l`	`y|n`	Whether the Primary Director port uses HTTPS.

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 summarized below:

Switch	Argument	Description
`start`	n/a	Starts the Engine. A second parameter starts the Engine with that Engine profile; this can also be `default` to download the default Engine profile and start with it.
`stop`	n/a	Stops the Engine.
`startfg`	n/a	Starts the Engine, and runs it in the foreground. Useful for monitoring or debugging purposes.

NOTE: If a machine is installed without a default route, the Unix Engine returns 0.0.0.0 as its IP address, which will cause communication issues with the Manager. You must ensure that a default route is set for Engines to operate properly. See page 17 for more information on properly configuring your network.

Configuring Engines

To change your Engine's settings (for example, to point them to a new Director), you change the configuration used by the Engine. Then, all Engines using that configuration will also change settings. To change an Engine configuration, go to the **Engine Configuration** page in the GridServer Administration Tool, select an Engine configuration, and change its values**.** Do not run the `configure.sh` script again.

WARNING When adding values to the **Environment Variables** box, it is possible to set variables, particularly `LD_LIBRARY_PATH`, that can cause an Engine to fail to start.

When new Engines are installed, they are assigned the default configuration for their platform (Windows, Linux, or Solaris) and will have the configuration defined for that profile.

You may assign groups of Engines to different configurations by creating a new configuration on the **Engine Configuration** page in the GridServer Administration Tool, and assigning it to Engines from the **Engine Daemon Admin** page. You could, for example, create a second configuration for a group of machines that points to a different Director and assign it to a subset of your existing Engines.

Configuring Dual-Homed Engines

In some network configurations, the PC running an Engine may have dual network interfaces, and the Engine may default to using the incorrect interface for its fileserver. To configure the Engine to use a different network interface, select the Engine Configuration that will be used by the Engine on the **Engine Configuration** page, and set the **Net Mask** value under the **File Server** heading to match the network range on which the Engine should run.

Running Multiple Engines from a Single Shared Installation Directory

Engines can be run from multiple hosts from a single shared installation mount point. All configuration information and logging is stored in the `profiles` directory for that Engine's hostname. Additional Engines are installed by simply starting the Engine from a new host; the `profiles` directory is automatically created and the Engine is assigned the default configuration.

If you run multiple Engine daemons from the same machine, you must run each daemon with a different UID.

Read-only Installation

Engines can be installed in a read-only directory, using a shared directory for data and work. In this configuration, DDT is enabled, but it uses a shared directory, and file updating is disabled. This enables you to use a file system such as IBM's GPFS for DDT.

To install in an installation directory that is read-only:

1. Configure the `profiles` directory in the **Engine Configuration** page to be in a writable location.
2. Disable file updating (it updates the root directory) in the **Resource Deployment** section of the **Manager Configuration** page on the **Manager** tab. You will need to use a shared mount point for resources and libraries; resources and libraries will have to be manually updated.
3. In the Engine configuration, change the data and work directories to be in writable locations.
4. In the Engine configuration, change the resource and library directories to the shared directories.

Multi-CPU Scheduling

Unix Engines provide the ability to tune scheduling for multi-CPU platforms. This section explains the basic theory of Engine distribution on multi-CPU machines, and how you can configure CPU scheduling to run an optimal number of Engines per machine.

GridServer can be configured to avoid conflicts between work and regular use of the machine. This is called adaptive scheduling, and can be configured to adapt to your computing environment. For example, you can specify a minimum and maximum CPU threshold. In the GridServer Administration tool, click the **Engine** tab, click the **Engine Configuration** page, and in a Unix profile, under the **Auto Mode** heading, select the "Start Engines when processor utilization is under" and "Stop Engines when processor utilization exceeds" check boxes. When non-Engine CPU utilization crosses below the minimum threshold for a given number of seconds, an Engine is allowed to run; when the maximum CPU usage on the machine is reached for a given number of seconds, the Engine exits and any Services it was processing are rescheduled.

With a multi-CPU machine, the processing power is best utilized if an Engine is run on each processor. Unix Engines have two types of CPU scheduling available to determine how Engines behave with multiprocessor systems: nonincremental and incremental scheduling.

Incremental Scheduling

Incremental scheduling is the method implemented by default in Unix Engines (and always with Windows Engines) to provide better scheduling of when Engines can run on multi-CPU computers. To configure incremental scheduling, go to the **Engine Configuration** page on the **Engine** tab in the GridServer Administration Tool. Under the **Auto Mode** heading, select the "Enable Incremental scheduling for multiprocessor systems" check box, and enter a time interval for how far apart Engines should be started.

With incremental scheduling, minimum CPU and maximum CPU utilization refers to each CPU. For example, if there is an Engine running on each CPU of a multi-CPU system, and the maximum CPU threshold is set at 80%, and a non-Engine program raises CPU utilization above 80% on that CPU, that

Engine will exit, and other Engines will continue to run until their CPU reaches the maximum utilization threshold. Also, an Engine would restart on that CPU when that CPU's utilization dropped below the minimum CPU utilization threshold.

The CPU scheduler takes the minimum and maximum per/CPU settings specified at Engine installation and normalizes the values relative to total system utilization. When these boundaries are crossed, an Engine is started or shut down and the boundaries are recalculated to reflect the change in running processes. This algorithm is used because, for example, a 50% total CPU load on an eight processor system is typically due to four processes each using 100% of an individual CPU, rather than sixteen processes each using 25% of a CPU.

The normalized values are calculated with the following assumptions:

1. For computing maximum thresholds, CPUs which do not have Engines running on them are taken to run at maximum capacity before usage encroaches onto a CPU being used by an Engine.
2. For computing minimum thresholds, CPUs which do not have Engines running on them are taken to be running at least the per/CPU maximum threshold.

The normalized utilization of the computer is calculated by the following formulas. The maximum normalized utilization (Unmax) equals:.

$$Unmax = \frac{Umax}{Ct} + \frac{Utot}{Ct}[Ct - Cr]$$

Where Umax = Per-CPU maximum (user specified), Utot = Maximum value for CPU utilization (always 100), Ct = Total number of CPUs, and Cr = Number of CPUs running Engines

The minimum normalized utilization (Unmin) equals:

The variables are the same as above, with the addition of Umin = per-CPU minimum.

$$Unmin = \frac{Umin}{Ct} + \frac{Umax}{Ct}[Ct - Cr - 1]$$

Nonincremental Scheduling

The simple form of CPU scheduling is called absolute, or nonincremental scheduling. In this method, minimum and maximum CPU utilization refers to the total system CPU utilization, and not individual CPU utilization. This total CPU utilization percentage is calculated by adding the CPU utilization for each CPU and dividing by the number of CPUs. For example, if a four-CPU computer has one CPU running at 50% utilization and the other three CPUs are idle, the total utilization for the computer is 12.5%.

With nonincremental scheduling, a minimum CPU and maximum CPU are configured, but they refer to the total utilization. Also, they simultaneously apply to all Engines. So if the maximum CPU threshold is set at 25% on a four-CPU machine and four Engines are running, and a non-Engine program pushes the utilization of one CPU to 100%, all four Engines will exit. Note that even if the other three CPUs are idle, their Engines will still exit. In this example, if the minimum CPU threshold was set at 5%, all four Engines would restart when total utilization was below 5%.

To disable incremental scheduling, go to the **Engine Configuration** page on the **Engine** tab in the GridServer Administration Tool. Under the **Miscellaneous** heading, clear the "Incrementally start and stop Engines" check box.

Chapter 6
Driver Installation

Introduction

The **Driver** is the component used to maintain a connection between the GridServer Manager and the client application. The GridServer SDK, available for Windows, Linux, and Solaris, contains several different Drivers. JDriver is used for Java applications; CPPDriver is used for C++ applications. PDriver is the parametric, or command-line interface Driver. The Windows SDK also contains the .NET Driver, which is used for .NET applications, and the COM Driver, used for COM applications.

This chapter contains information on installing the GridServer SDK, which includes each type of Driver.

Note: Solaris systems require GNU tar to unarchive the GridServer SDK.

GridServer SDK Installation

The following procedure explains the basics for installing the GridServer SDK. There are additional steps for each Driver. Read both the SDK instructions and additional instructions for each Driver you are using after the SDK installation.

To install the GridServer SDK:

1. In the GridServer Administration Tool, click the **Driver** tab, click **SDK Download**, and click the SDK for your platform to download it.
2. Unzip or unarchive the SDK.

There is a `driver.properties` file that can be downloaded from the **SDK Download** page; this should be copied to the `config` directory of the GridServer SDK. You may need to edit the Manager settings so that the Driver will point to your local Managers.

3. Read the additional steps, below, for each Driver you are using.

The Java Driver (JDriver)

The Java Driver, also known as JDriver, consists of a JAR file used with your Java application code.

Perform the following installation steps after installing the SDK if you plan to use JDriver:

1. Ensure you have the Java 2 SDK installed (also commonly referred to as the JDK.) You can download it from `java.sun.com`.
2. Define an environment variable `JAVA_HOME` that contains the location of the JDK.

Each of the Java examples in the `examples` directory of the GridServer SDK includes an `env`, `build`, and `run` script, which demonstrate how to properly set classpaths and environment variables to run a Java application using JDriver.

To use the Driver add the `DSDriver.jar` file and the `config` directory of the GridServer SDK to your classpath when running your application.

The C++ Driver (CPPDriver)

The C++ Driver, also known as CPPDriver, consists of DLLs and include files that are linked with your GridServer application. For Windows, VC6, VC7, and VC7.1 libraries are available; for Unix, GCC2, GCC3, and Solaris are supported.

Perform the following installation steps after installing the SDK if you plan to use the C++ Driver:

1. Set the environment variable `DSDRIVER_DIR` to the path of the `config` directory in the SDK.
2. For Windows, add the directory `cppdriver/bin/vc6` (or `vc7` or `vc71`, depending on which you use) to the `PATH` environment variable.
3. For Unix, set the `LD_LIBRARY_PATH` environment variable to include the applicable directory in the `lib` directory.

The Parametric Job Driver (PDriver)

PDriver, or the Parametric Job Driver, is a Driver that can execute command-line programs as a parallel processing job using the GridServer environment. This enables you to take a single program, run it on several Engines, and return the results to a central location, without writing any new code.

Perform the following installation steps after installing the SDK if you plan to use PDriver:

1. In the `pdriver` directory of the SDK, there is a `setenv.bat` or `setenv.sh` script, which properly sets the path, `DSDRIVER_DIR` and `DS_SDK_DIR` environment variables for Drivers. This script should be run before you use the Driver. The best method of doing this is running it from a login script. Note that for unix you should run the `setenv` script from the `pdriver` directory with a `source setenv.sh`.

Running PDriver on Windows Engines

The Windows `Pdriver.dll` was compiled using Visual C++ 6, so Windows Engines must use the VC 6.0 version of the C++ bridge. PDriver uses the default library path in the Engine Configuration, typically `./resources/win32/lib`, so you must make sure that `DSBridgeVC6.dll` is the only bridge in that directory. If you have other Services that require other bridges, you must use Grid Libraries.

Running PDriver with Linux

There are a number of different compilers available for the various Linux distributions, and not all platforms support all compilers. Because of this, each Linux platform supported has a compiled `libpdriver.so` file. PDriver uses the default library path in the Engine. To properly deploy PDriver, the proper C++ bridge library and PDriver library must be deployed in that path. For example, on an Engine that runs on a platform that only supports GCC 3.0, you would deploy `libpdriverGCC3.so` and `libdsBridgeGCC3.so` to `resources/linux/lib/gcc3`, and set the Default Library Path to `./resources/linux/lib/gcc3`.

Using PDriver

To run a PDriver script file, invoke the PDriver binary and pass the PDriver script file as the first argument. For example, on Unix systems:

```
pdriver examples/example.pds
```

Running `pdriver` or `PDriver.exe` without a PDS argument will provide a usage message.

The `examples` directory contains several example PDriver script files.

The .NET Driver

The .NET Driver consists of an assembly that includes classes used to create and manage Services from .NET. The .NET Driver is only available for Windows.

Perform the following installation steps after installing the SDK if you plan to use the .NET Driver:

1. The `GridServerNETClient.dll` is located in the `NETDriver` directory. You need to link this assembly to your application.
2. Set the environment variable `DSDRIVER_DIR` to the path of the `config` directory in the SDK.

COM Driver

The COM Driver enables an application using the Component Object Model architecture to access GridServer, enabling distributed parallel execution of the application on a Grid of Engines. The Driver includes an example, an Option Evaluation spreadsheet in Excel that uses GridServer for its calculations. The COM Driver is only available for Windows.

Perform the following installation steps after installing the SDK if you plan to use the COM Driver:

1. In the `COMDriver` directory of the SDK, double-click the `DSCOMDriverInstall.exe` to start the installer.
2. After the InstallShield welcome screen, click **Next** to continue.
3. Select a location for the COM Driver by clicking the **Browse** button. The default location is `C:\DataSynapse\DSCOMDriver`. Click **Next** to continue.
4. Enter the address of your primary and secondary Directors, in the form `hostname:port`. For example, `http(s)://server1.mycompany.com:8000`. You can proceed without filling in any values, but you will need to later specify your Directors by editing the `driver.properties` file. Click **Next** to continue.
5. Click **Finish**.

Driver Configuration

Drivers are configured with a `driver.properties` file that can be downloaded from the **SDK Download** page; this should be copied to the `config` directory of the GridServer SDK. When your Manager is installed, this `driver.properties` file is preconfigured with the primary and secondary Director. You can edit the file to change any default values, for example to add a username and password for authentication, change the primary and secondary Director, change the Broker Timeout, and so on.

The `driver.properties` file can be moved to another directory. For Java, the new directory must be added to your classpath. For C++, .NET, and PDriver, the `DSDRIVER_DIR` environment variable must be set to the location of this directory. For PDriver, edit the `setenv.bat` or `setenv.sh` script to set the `DSDRIVER_DIR` environment variable.

Any properties added to the `driver.properties` file using characters `#`, `!`,`# =`, `\`, and `:` should be escaped with a preceding backslash.

When a property is a hostname or directory, there is special handling of the backslash character on Windows Drivers. The first backslash, indicating the root directory, is translated into the current Windows drive. Other backslashes are ignored. Forward slashes are translated into backslashes.

For example, to set a directory to `c:\sdk\log`, use `/sdk/log` in the `driver.properties` file. To use a UNC path such as `\\homer\job1-dir`, use `//homer/job1-dir` in the `driver.properties` file.

You can also set Driver properties programatically with the API. When this is done, the `driver.properties` file is not needed. At minimum, `DSPrimaryDirector` and `DSSecondaryDirector` should be set (or set them both to the same hostname when using a single Director) and all other properties will use the default settings.

Configuring Dual-Homed Drivers

In some network configurations, the PC running a Driver may have dual network interfaces, and the Driver may default to using the incorrect interface. To configure the Driver to use a different network interface, set the `DSLocalIPAddress` property to the IP number of the correct interface. For example:

```
DSLocalIPAddress=192.168.12.1
```

Multiple Driver Instances

If you enable Direct Data Transfer and you write a script that instantiates multiple Drivers from the same `driver.properties` file with the same port number, the first Driver will open a web server listening to the defined socket. Subsequent Drivers will not open another web server as long as the first Service is running, but will be able to continue running by using the first Service's server for direct data. However when the first Service completes, its server will be terminated, causing subsequent Services to fail.

You can avoid this by writing a shell script to create Services, each with their own Driver running from its own Java VM. Your script must provide a different port number for the `DSWebserverPort` property normally set in the `driver.properties` file. To write a shell script for this situation, you could remove the `DSWebserverPort` property from the `driver.properties` file and assign a unique port number for each iteration.

Also, more than one Driver cannot share the same directory. If you script multiple Drivers to run from the same `driver.properties` file, you will need to set a different `DSCacheRootDirectory` `DSLogDir`, and `DSWebServerDir` properties for each Driver.

Chapter 7
Verifying your Installation

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

Submitting a Service Test

A Service Test can run a Linpack benchmark that can be used 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, click the **Services** tab, then click **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 will take longer to complete.
4. You will be brought to the **Services Admin** page on the **Services** tab, where you can monitor the Service's progress.

Monitoring a Service Test

Once the Service Test is submitted, you can monitor its progress to ensure that your Engine is properly taking tasks and returning results. While the Service Test is running, there are several other pages you can view for more information:

- Click the **Manager** tab, then click **Broker Admin**. A list of available Brokers is displayed, along with the number of busy, idle, and total Engines reporting to that Broker.
- Click the **Manager** tab, then click **Broker Monitor**. This opens a new screen with four interactive graphs. These include an Engine Monitor, Service Monitor, Task Monitor, and System Monitor, which displays memory and thread activity. You can also use the graph icon in the Administration Tool to open the Broker Monitor.
- Click the **Engine** tab, then click **Engine Admin** for a list of all Engines and their status.

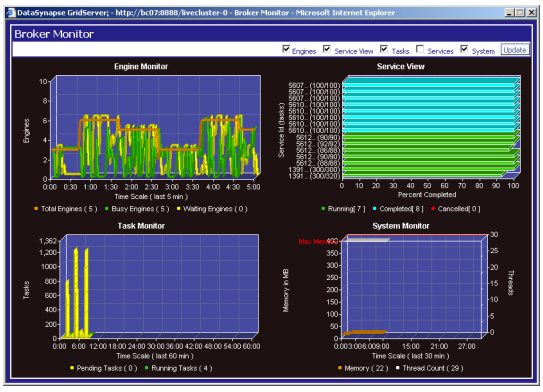


FIGURE 7-1: The Broker Monitor.

Troubleshooting

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

Chapter 8
Uninstalling GridServer

This chapter explains how to uninstall GridServer. The Engine and Manager uninstallation are separate procedures as explained below, for Windows and Unix operating systems.

Uninstalling GridServer on Windows Systems

Uninstalling an Engine

To uninstall an Engine on a Windows system:

1. If you've previously engaged the Engine manually, right-click the DataSynapse icon in the Task Tray and click **Stop**

   The DataSynapse Icon will turn red.

2. Right-click the red DataSynapse icon in the Task Tray and click **Exit**.From the **Services** menu (under **Administrative Tools** in the Windows **Control Panel**) click **DataSynapse Engine** to stop the application as a Service.
3. From the Windows **Control Panel**, under **Add/Remove Programs**, remove the **DataSynapse Engine**. The InstallShield Wizard guides you through the uninstall.

FIGURE 8-1: Stopping the Engine from the Engine Menu

FIGURE 8-2: Exiting the Engine from the Engine Menu

When the uninstall is finished, delete the remaining Engine directories from your disk. (Recall that the default location for these files is under `C:\Program Files\DataSynapse`.)

Manual Uninstall

DataSynapse Engines can be removed using silent installation scripts similar to those described in "Network Installation" on page 33. However, instead of the `install.bat` script, you use the `uninstall.bat` script.

To uninstall an Engine manually:

1. Change directories to the shared network installation directory containing the `uninstall.bat`. If you don't still have this directory from your installation as described in "Network Installation" on page 33, you can re-copy the contents of `[GS Manager Root]/webapps/livecluster/WEB-INF/engineInstall/win32/network` to a shared directory.
2. Stop the Engine. If you want to script this, you could write a batch file that contains the command `net stop "DataSynapse Engine"` before the uninstall command.
3. Run `uninstall.bat` from the shared network installation directory.

Uninstalling a Manager

To uninstall a Manager with Resin on a Windows system:

1. If you installed the Manager as a Service, click the **Control Panel** window in the **Start** menu, click **Administrative Tools**, then click **Services**. Click **DataSynapse Manager** and click the Stop icon to stop the application.

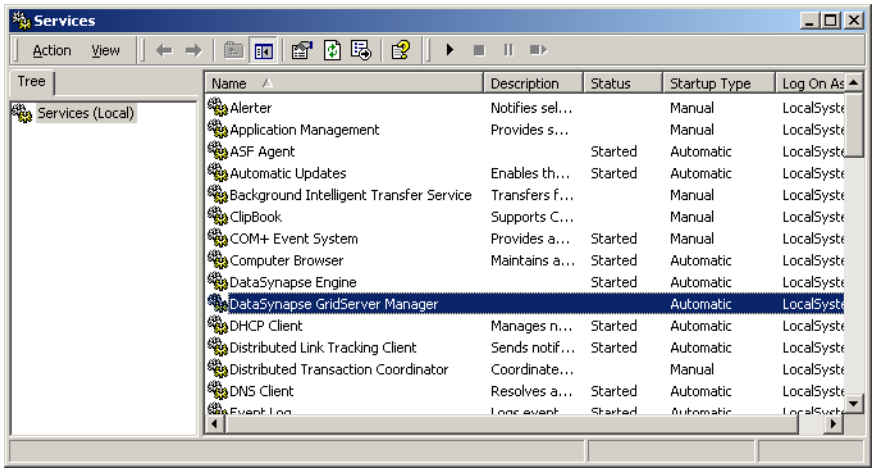


FIGURE 8-3: Stopping a Manager installed as a Windows Service.

2. Open a command line, go to the GridServer Manager root directory, and issue a `server remove` *name* command to uninstall the Server named *name*. Omit *name* if you did not name your Manager during installation.
3. In the `[GS Manager Root]/webapps` directory, remove the `livecluster` directory and its contents. You can also remove the entire `[GS Manager Root]` directory containing GridServer and the application server.

Uninstalling GridServer on Unix Systems

To remove GridServer Managers or Engines on a Unix system:

1. Remove any modifications you made to startup files, typically located in `/etc/rc.d` directories.
2. Remove all files in the installation directory.

Chapter 9
Upgrading GridServer

Introduction

This chapter explains how to upgrade GridServer. It contains a GridServer Upgrade Checklist, which explains what to do differently before, during, and after your new GridServer installation to ensure old values are carried over to the new installation, when possible. Also described are any changes in requirements for running GridServer 4.2, and a list of API changes between GridServer 4.0, 4.1, and 4.2.

Requirements Changes

Please see the system requirements section on page 11 of the *GridServer Installation Guide* to ensure that your software and hardware meet the minimum system requirements. Below is a list of some requirements that have changed since GridServer 3.2. However, make sure to review the full list of requirements.

Memory Requirements

The minimum RAM required for running the GridServer Engine is now 512 MB. The minimum RAM required for running the GridServer Manager is 768 MB; 1 GB is recommended.

Operating Systems

Operating systems no longer supported in GridServer 4.2 include:

- Windows 95
- Windows 98
- Windows ME
- Windows NT 4.0
- Solaris 2.6
- Solaris 7
- Red Hat Linux 7.0

Java Requirements

The required Java Virtual Machine for Managers and Drivers is now 1.4.2_05. You cannot use builds 01-04 of 1.4.2; you must upgrade to build 05 or newer.

.NET Framework Requirements

If you use the .NET Driver, GridServer requires the .NET Framework 1.1 to be installed on GridServer Engine machines. Note that the 1.0 Framework is no longer supported. Also, to use SSL with the .NET Driver, you need .NET Framework 1.1 SP1.

Other Libraries

The libz library, which is used by the Linux CPPDriver has been updated from 1.1.3 to 1.2.1.

The libcurl library replaces libghttp in the Solaris and Linux versions of CPPDriver.

Other Changes

DBGDS is no longer supported for Windows Engine Debugging. See “Running Engines in Console Mode” on page 36 for more information on running Engines in console mode to view logging information.

GridServer 4.2 Upgrade checklist

Settings from your previous version of GridServer can be migrated 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 GridServer 4.2 in the same location. Later in the install, you will be asked for a path to the renamed old installation so settings can be copied.

Note that Manager configuration values, such as backlog, max queue, log level, timeout seconds, and so on, do not migrate to the new version during a Manager upgrade. However, an `upgrade.log` will be generated that lists all configuration values that have been changed, so that you can restore your settings.

Prior to upgrade

☐ If you used .NET in a previous installation, complete the relevant steps in the section entitled “.NET Upgrade” on page 57.

☐ If you have previously used the system classpath in your Engine configuration and deployed the `DSEngine.jar` file to Engines, remove the deployed `DSEngine.jar` and let Engines update before upgrading the Manager. After the Manager upgrade, deploy the new `DSEngine.jar` to Engines.

If you have already upgraded without deleting the old `DSEngine.jar`, you must turn off Engine version check by going to the **Manager** tab of the GridServer Administration tool, selecting the **Manager Configuration** page, selecting the **Admin** section, and setting **Allow Engine Version Mismatch** to True, which will enable Engines to log in and download the new `DSEngine.jar`. Once they have, the version check can be turned on by setting **Allow Engine Version Mismatch** to false.

After stopping the Manager

☐ Rename your Manager directory if you plan on installing GridServer 4.2 in the same location. Do not install GridServer 4.2 over the existing installation.

☐ If you are running GridServer as a Windows service, you should remove the service prior to upgrading.

During installation

☐ For **Installation Type**, choose **Custom, Manager Upgrade**.

☐ Under **Previous Base Directory**, enter the `livecluster` directory of your previous installation, such as `/datasynapse.last/webapps/livecluster`.

☐ If you have installed a previous version of GridServer with an alternate base directory and you upgrade to a new version, you must use the same directories for the new installation.

Also, when upgrading from an alternate base directory to an alternate base directory, you can enter only the base directory, instead of the livecluster directory, such as `/datasynapse.alt` instead of `/datasynpapse.alt/webapps/livecluster`.

☐ If you made changes to the default database values in the previous installation, you will need to re-enter them, as all database values will be reset to their default value. You should confirm all database values, especally when running an external reporting database. All other values will be carried over.

After Installation

☐ Check the `upgrade.log` file in the `livecluster/WEB-INF/log` directory. Any changes made from the default in the prior version will be logged. All values are restored to default on an upgrade, so you may wish to restore these changes if they still apply.

☐ The GridServer distribution requires a license, which must be obtained from the DataSynapse support team. Contact your account representative or email support directly to obtain a license. If you already have a license from your previous installation of GridServer, it will work in GridServer 4.2, but you will need to move it to your new installation.

☐ The .NET Driver now requires a license that specifically enables that Driver type. Existing licenses will not allow the .NET Driver to connect. Contact your account representative or email support to obtain a new license.

> **NOTE:** The license installation procedure changed in GridServer 4.1; licenses are now installed from the **License Information** page in the GridServer Administration Tool, and no longer require a Manager restart.
>
> For more information on installing licenses, see "Installing Manager Licenses" on page 29 of the *GridServer Installation Guide*.

☐ GridServer no longer supports .NET Framework 1.0. Running executables linked with .NET Driver against Framework 1.0 is not supported, and running Engine side code against Framework 1.0 is not recommended, although it is supported for this release. Since there are no binary incompatibilities between 1.0 and 1.1 for public types in the Framework, it will not be necessary to rebuild Driver-side or Engine-side code. However, some user code may rely on portions of the Framework that behave differently in version 1.1. See `http://www.gotdotnet.com/team/changeinfo/default.aspx` for more information.

☐ If you previously registered any C++ Service Types on the **Service Type Registry** page and they had a value for the **Definition** field, this will not carry over to the new installation. To remedy this, open the Service Type and enter a new description as needed.

☐ GridServer is no longer installed with a reporting database by default. You should configure GridServer to use an external reporting database, if needed. For more details, see Appendix B, "Database Configuration" on page 59.

☐ To upgrade an external reporting database schema from a previous version of GridServer, run the upgradedb.sh or upgradedb.bat script located in `[GS Manager Root]/webapps/livecluster/WEB-INF/etc/db`.

- The Engine file cleaner will now only clean directories beginning with `ds-`, so files that existed before an upgrade, such as the Engine log files, will not be automatically cleaned. You can simply remove these files later if needed.
- The CPU utilization field in the reporting database is now in percent utilization; previously this was reported as MFLOPS.
- The .NET Driver no longer backgrounds its threads when there are inactive but non-destroyed Services. This is a change from the 4.1 behavior, in which the Driver threads would all go into the background when there were no active tasks. If you do not call `System.Environment.Exit` explicitly, you will now need to make sure you call `Service.Destroy` or `Service.DestroyWhenInactive` to properly clean up your Services so that the Driver knows when to background the collection thread.
- Data from active GridCache regions that is stored on disk will not be accessible after upgrading to GridServer 4.2.
- Broker-level resources are no longer supported in GridServer 4.2. You will need to deploy any existing Broker-level resources using Grid Libraries or legacy resource deployment.
- In Grid Libraries, a `pathelement` element can no longer be set to an absolute path; only relative paths are supported.
- The `BrokerAdmin.setEngineWeight` and `BrokerAdmin.setDriverWeight` methods changes from float to double, which may cause exceptions when using SOAP stubs. To avoid this issue, regenerate the stubs for clients that use these methods.
- The McKoi database is no longer supported as a reporting database.

Engine Upgrade

- If there are multiple Unix Engines running from the same shared directory, the Engines will not auto-update to the latest version. You must manually install the new Engine once into the shared installation directory, which will cause all Engines to run the latest version.
- Engine configurations whose `LIB_DIR` specify absolute paths containing .Net assemblies will not work after upgrading. They should be changed to relative paths.

Driver Upgrade

- You should 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, they should be relinked with the new libraries included in the SDK.
- On the Driver side, you should link your code with the `dsUtil` library; Engine-side Service code should link with the `dsUtil` library if it uses any exception classes.
- If you have existing .NET applications, you must change the reference to the new `GridServerNETClient.dll` and rebuild.

- In any Makefiles, you should link with the 1.2.1 version of the libz library; and replace the libghttp library with the libcurl library for the Solaris and Linux versions of CPPDriver.
- In Linux, there are different SDK downloads for gcc2 and gcc3 versions of CPPDriver. This changes the directory structure of CPPDriver. For example, the `$SDK_DIR/cppdriver/lib/gcc3` directory is now simply `$SDK_DIR/cppdriver/lib` in the gcc3 SDK download.

Default Engine Configuration Changes

The following changes take place in the default Engine properties:

Property	Default Value
New properties	
CPU Recording Interval	5
Grid Library Path	`./resources/gridlib`
Changed properties	
Mflops Report Frequency	1 (was 0)
Deprecated	
Code Version	

Default Manager Configuration Changes

The following changes have taken place in the default Manager configuration:

Property	Default Value
managerKeyStore	`$DSWebappDir$/WEB-INF/certs/manager.keystore`
keyStorePassword	
ManagerKeyStorePassword	
defaultAffinity	`ServiceAffinity`
TokenTimeout	600 (was 100)
checksPerMinute	2.0 (was 4.0)
maxMillisPerHeartbeat	120000 (was 30000)
TimeoutFactor	2 (was 3.0)
ScheduleBusy	false (was true)

Property	Default Value
fileUpdateLocation (additional class GridLibraryDeploymentLocation)	serverDir=`deploy/resources/gridlib` clientDir=`resources/gridlib` Platform=`all` recursive=`false`

Incompatibilities due to API changes

In GridServer 4.2

- Setters/getters for individual `JobOptions` properties , which have been deprecated since GridSever 3.1.0, have had their documentation removed. You should use the generic set/getProperty methods instead. These methods will be removed in the next major GridServer release.
- LiveCache has been removed from the API for .NET. (It was previously deprecated.) GridCache should be used instead.
- In CPPDriver, the deprecated admin API in the `admin_old` directory has been removed.

In GridServer 4.1

Code Versioning has been deprecated. Grid Libraries should be used instead. For information on migration, see “Code Versioning Deprecation” on page 53 of the *GridServer Administration Guide*.

PDS Incompatibility

The output of backtick commands is no longer explicitly converted to a range type. Backtick command output must now be passed through the split() command in order to be properly used within task blocks, and by `sizeof`, `for`, and `foreach`.

Mixed-Version Deployments

Mixed-version deployments are grids that consist of both GridServer 4.1 and GridServer 4.2 Brokers. This is an allowed configuration, as a convenience for large Grids that cannot upgrade all Brokers at once. There are several issues that must be addressed when running in this mode, which are outlined below.

To run a mixed-version Grid:

1. You must have GridServer 4.2 Directors. While it is possible to run GridServer 4.2 Directors with GridServer 4.1 Brokers, you can’t do the reverse.
2. Brokers must be patched to GridServer 4.1.0.24.
3. You have to disable client version checking for Engines, Drivers, and Brokers on the **Manager Configuration** page in the **Admin** section.
4. You must use the 4.1-compatible Engine balancer. This is set on the **Manager Configuration** page, in the **Engines and Clients** section, with the **Engine Balancer** parameter.

5. We recommend hard-partitioning Engines so that some will go only to GridServer 4.1 Brokers and others only to GridServer 4.2 Brokers. This can be done with Broker routing or home/shared Brokers. If you don’t do this, Engines will perform an additional restart if it was on a GridServer 4.2 Broker and is routed to a GridServer 4.1 Broker, and vice versa.
6. You must set up Driver routing rules either by Broker routing or using Driver profiles to prevent GridServer 4.1 Drivers from logging in to GridServer 4.2 Brokers.
7. You must have both versions of the C++ Bridge Grid Libraries deployed via the Primary Director. If you upgraded the Director, the GridServer 4.1 bridges will already be there, but if the Director is a new installation you will need to copy them from your GridServer 4.1 Manager. The Engines will automatically choose the proper library based on the Broker version.

   Because both versions of the C++ Bridges are used, gridlibs must also be used on a GridServer 4.1 Broker if the code is native, rather than using the standard resource deployment.

8. In GridServer 4.2, a routing optimization was added, in that the Director does not send the login message to the Broker; it immediately returns a message to the Engine specifying the Broker which it should log in to. This provides faster processing of Engine logins since there is no Director-to-Broker communication involved. However, this optimization only works for GridServer 4.2 Brokers; Engines routed to GridServer 4.1 Brokers will still wait on the Director while the login message is processed by the Broker.
9. A mixed grid doesn't work with Engines running on a shared NFS-mounted directory.

Appendix A
Using DSHook

Windows GridServer Engines typically run as a service, using the Local System account to monitor local keyboard and UI activity and determine when a computer is idle. However, your installation environment may make it impossible to install Engines using the Local System account, and installation of an Engine as a service using another account would prevent it from monitoring system activity. To install an Engine in this situation, you can install the included program `DSHook.exe` as a service, using the Local System account. DSHook only counts local keyboard and UI activity, so it is not considered to be a security risk. Engines are installed under another username, and simply communicate with DSHook to monitor the system.

Installing DSHook

The following instructions describe how to install DSHook as a service by hand, using the Registry Editor and the `instsrv.exe` program included in the Windows Resource Kit. You can also use other programs to install the service automatically and script the editing of the registry. For example, `sc.exe`, also included in the Windows Resource Kit, can be used to automate several of the below steps, and can also be used to create services on a remote computer.

Note that you will need the `instsrv` program, which is included with the Windows NT/2000/XP Resource Kit for the installation.

To install DSHook:

1. Log onto the machine as a user with administrative privileges.
2. Open a command prompt and change to the Engine directory, typically
   `C:\Program Files\DataSynapse\Engine`.
3. Type:
   ```
   instsrv DSHook C:\Program Files\DataSynapse\Engine\srvany.exe
   ```
   You can replace `DSHook` with another name for the service, if desired. Replace the path to `srvany.exe` if your Engine is installed in a different directory.
4. Type `regedit` to open the Registry Editor.
5. Locate the following key:
   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DSHook`
6. From the **Edit** menu, click **New**, click **Key**, and name the new key **Parameters**.
7. Highlight the **Parameters** key.
8. From the **Edit** menu, click **New**, click **String Value**, and name the new value **Application**.
9. From the **Edit** menu, click **Modify**, and type in the path name for the Engine directory, including the drive letter, plus the name `DSHook.exe`. Enclose the full path in quotation marks.
10. From the **Edit** menu, click **New**, click **String Value**, and name the new value **AppDirectory**.
11. From the **Edit** menu, click **Modify**, and type in the path name for the Engine directory, including the drive letter. Don't enclose the path in quotation marks.

12. Close the Registry Editor.
13. Ensure that the file `workerhook.dll` is in the Engine directory with `srvany.exe`. The `workerhook.dll` file must have read/write/execute permissions for all domain users. If your policies therefore require you to move `workerhook.dll` into a shared system area, such as `C:\WINNT\SYSTEM32`, you must ensure that only one copy of `workerhook.dll` exists (removing it from `C:\PROGRAM FILES\DATASYNAPSE\ENGINE` if necessary).
14. From the **Start** menu, open **Control Panels**, then **Administrative Tools**, then **Services**.
15. Right-click the DSHook service, click **Properties**, then click the **Log On** tab. Ensure the service is using the Local System account, and that **Allow service to interact with desktop** is selected.
16. Right-click the DSHook service and click **Start** to start the service.

Installing Engines using DSHook

After installing DSHook, you can then install Engines that log in with any other user account. Note that Engines will require read access to `Windows` and `System32` directories, and must also have read access to the registry.

You can also install DSHook on an existing Engine. After installation, ensure that the Engine's Service is updated properly so that the Engine has no Desktop access.

Appendix B
Database Configuration

The GridServer Manager includes the ability to use a database to store tables for Services, tasks, users, statistics, and other records. The GridServer Manager includes HSQLDB, which can be used as a database for reporting. However, you can also configure GridServer to use an enterprise-grade database from another vendor.

Note: If a Manager has already been running against a different third-party database and the contents of the database have not been migrated to the new database, all data will be lost. It is recommended that you establish the database you plan on running against prior to Manager initialization.

Installing Standalone HSQLDB

GridServer includes a sample HSQLDB reporting database, primarily intended for demonstrating the use of a reporting database. It is located at `[GS Manager Root]/webapps/livecluster/WEB-INF/etc/db/hsqldb`.

Its default port is 5000, and it may be changed in the `server.properties` file. The username and password are set in the `reporting.script`, and are `ADMIN`/`ADMIN` by default. The database is started by running the `run.bat` or `run.sh` script.

To use HSQLDB as your test reporting database, copy the `[GS Manager Root]/webapps/livecluster/WEB-INF/etc/db/hsqldb` directory to an appropriate application location, start the database, and follow the database configuration instructions below. The steps for setting the `CLASSPATH` and running the `createdb` script can be skipped.

Database Configuration

There are essentially three steps to configuring the database. First, you install your Manager; second, configure the database; third, configure the Manager to use the database. You will also need to restart the Manager after configuring it for a new database.

First Step: Install the Manager:

1. Install and configure your GridServer Manager, if you haven't installed it already. You shouldn't change any of the database settings during installation. Restart the Manager when you are finished with the installation.
2. Ensure that the `tables.<db>.sql` file needed by the `createdb` script is located in the `[GS Manager Root]/webapps/livecluster/WEB-INF/etc/db` directory. Databases supported at this time are HSQLDB, Microsoft SQL server, Oracle, Sybase, and DB2.

Second Step: Configure the database (these steps can be skipped for HSQLDB):

1. Set the `CLASSPATH` on the database machine to include the database driver JAR files applicable to your database resides. For example, to create an MSSQL2k db in a unix shell:

```
export
CLASSPATH=[JDBCDriverDir]/msbase.jar:[JDBCDriverDir]/mssqlserver.jar:[JDBCDriverDir]/msutil.
jar
```

replacing `[JDBCDriverDir]` with the actual path to the directory containing your driver JAR files.

2. Run the `createdb` script provided in `[GS Manager Root]/webapps/livecluster/WEB-INF/etc/db` directory on your Manager. You must pass the name of the appropriate properties file as an argument, specify the one which corresponds to your database. Before you run the script, change the property file settings to match those of your database.

Third Step: Configure the Manager:

1. Log in to the GridServer Administration Tool with an account that has configure access.
2. Click the **Manager** tab, click **Manager Configuration**, then click **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
HSQLDB	
URL	jdbc:hsqldb:hsql://*hostname*:5000/reporting
Driver	org.hsqldb.jdbcDriver
Transaction Isolation	TRANSACTION_READ_UNCOMMITED
Database Type	`HSQL`
Oracle 9i:	
URL	jdbc:oracle:thin:@*hostname*:1521:*database*
Driver	oracle.jdbc.OracleDriver
Transaction Isolation	`TRANSACTION_READ_COMMITTED` (The default value is `TRANSACTION_SERIALIZABLE`, which does not work properly with Oracle.)
Database Type	oracle9i
Sybase Adaptive Server Enterprise 5.5:	
URL	jdbc:sybase:Tds:*hostname*:4100/*database*
Driver	com.sybase.jdbc2.jdbc.SybDriver
Database Type	sybase12
IBM DB2 8.0.1 (and other versions):	
URL	jdbc:db2://*hostname*:50000/*database*
Driver	com.ibm.db2.jcc.DB2Driver
Database Type	db2udbv8

Property	Value
v7	db2udbv7
v8 for zOS	db2zosv8
Microsoft SQL Server 2000 (8.0):	
URL	jdbc:microsoft:sqlserver://*hostname*:1433;DatabaseName=*database*;SelectMethod=Cursor (SelectMethod=Cursor is required)
Driver	com.microsoft.jdbc.sqlserver.SQLServerDriver
Database Type	mssqls2k

4. Copy the corresponding JDBC driver's JAR file into the Manager's `[GS Manager Root]/webapps/livecluster/WEB-INF/lib` directory.
5. Restart the GridServer Manager.

Third Party Database Known Issues

Microsoft SQL Server

Setting the `SelectMethod` property to "`Direct`" will result in errors. `SelectMethod` must be set to "`Cursor`" to prevent direct query connection cloning from conflicting with the GridServer Manager's connection pooling scheme.

The `SelectMethod=cursor` property is only required when using the Type 4 JDBC drivers provided by Microsoft. These are the only drivers that have been tested.

Oracle

GridServer has only been tested against the Oracle 9i server with Type 4 drivers. The "Thin" driver for Oracle 8 is a type 4 driver and should work as well. Oracle's documentation suggests that their OCI (Type 2) drivers will not properly support Timestamp data types, which are used heavily by the GridServer Manager.

Sybase

By default, Sybase does not allow null values in string columns (like `VARCHAR`, `TEXT`, and so on.) You can change this default value for a database using `isql` by issuing the following command:

```
sp_dboption databasename, "allow nulls by default", true
```

In this example, `databasename` is your database name. This command must be executed before creating the DB schema.

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

W