

Getting Started with vFabric Hyperic

VMware vFabric Hyperic 4.6.6

VMware vFabric Suite 5.2

VMware vFabric Suite 5.1

VMware vFabric Cloud Application Platform 5.0

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see <http://www.vmware.com/support/pubs>.

EN-000925

vmware®

Table of Contents

1	About Getting Started with vFabric Hyperic	5
	Intended Audience	5
2	vFabric Hyperic QuickStart Installation	6
3	Key Facts for Hyperic Installers	7
	Hyperic Security Features and Recommendations	7
	About SSL in Hyperic	9
	About Agent Configuration	11
	About the Agent Launcher and Agent Startup	13
	About Agent - Server Communication	15
	Agent Directory Structure	17
	About Sizing Profiles in vFabric Hyperic	18
4	Hyperic Installation and Startup Process	20
	Step 1 – Obtain Hyperic Installer	20
	Step 2 – Set Up Hyperic Database	20
	Step 3 - Set Up Hyperic Server	22
	Step 4 - Set Up the Hyperic Agent	23
	Step 5 - Next Steps for New Installations	27
5	Hyperic Upgrade Processes	28
	Hyperic 4.6 Upgrade Options	28
	Upgrade Hyperic Server Only	28
	Upgrade Server and Agent with Hyperic-Managed Keystores	28
	Upgrade Server and Agent with User-Managed Keystores	29

6	Step-by-Step Procedures	30
	Select and Download an Installer	31
	Configure JREs for Hyperic Components	33
	Configure SSL Options	35
	Set Up MySQL	37
	Set Up Oracle	41
	Set Up PostgreSQL	48
	Run Hyperic Installer	53
	Install an Agent-Only Package	56
	Run Windows MSI Installer	57
	Install Hyperic Server RPM	72
	Install Hyperic Agent RPM	79
	Set Up Agent Interactively	82
	Set Up Agent in Properties File	84
	Encrypt Agent Property Value	99
	Install Hyperic Agents in Volume	100
	Upgrade Hyperic Agent	103
	Upgrade Hyperic Server	106
	Uninstalling an Agent	110
	Change vFabric Hyperic Server Sizing Profile	111
	Configure Unidirectional Agent - Server Communication	112
	Install or Configure vFabric Hyperic License	114
7	Troubleshoot Agent and Server Problems	116
	Looking for Clues	117
	Hyperic Server Problems	119
	Large Event Table	119
	Agent Startup or Connection Problems	120
	Invalid or Unknown Availability	122
	Slow User Interface	124
	Warning Messages in the Agent Log	124

- [About Getting Started with vFabric Hyperic \(see page 5\)](#)
- [vFabric Hyperic QuickStart Installation \(see page 6\)](#)
- [Key Facts for Hyperic Installers \(see page 7\)](#)
- [Hyperic Installation and Startup Process \(see page 20\)](#)
- [Hyperic Upgrade Processes \(see page 28\)](#)
- [Step-by-Step Procedures \(see page 30\)](#)
- [Troubleshoot Agent and Server Problems \(see page 116\)](#)

About Getting Started with vFabric Hyperic


1

Getting Started with vFabric Hyperic provides procedures for installing vFabric Hyperic components, including setting up the Hyperic database and installing the Hyperic Server and Hyperic Agents.

Intended Audience

Getting Started with vFabric Hyperic is intended for operations personnel who set up and support the core Hyperic infrastructure.

vFabric Hyperic QuickStart Installation

 **Note:** If you obtain VMware® vFabric™ Hyperic® as part of a vFabric Suite (previously known as "vFabric Platform"), first complete the license activation and installation procedures in *Getting Started with vFabric Suite*. Then follow procedures in this document to set up your environment for vFabric Hyperic and complete any remaining Hyperic-specific installation and configuration tasks.

To evaluate Hyperic components, you can get up and running quickly by installing the Hyperic Server and a Hyperic Agent on the same host, using Hyperic's built-in database.

For comprehensive installation information, see [Hyperic Installation and Startup Process \(see page 20\)](#).

Procedure

1. Verify Supported Configurations and System Requirements for the Hyperic Server and Hyperic Agent.
2. [Select and Download an Installer \(see page 31\)](#). Choose a "full" installer.
3. Install the components as described in [Run Hyperic Installer \(see page 53\)](#).
4. For post-installation tips, review [Hyperic Installation and Startup Process \(see page 20\)](#).

Key Facts for Hyperic Installers

- [Hyperic Security Features and Recommendations \(see page 7\)](#)
- [About SSL in Hyperic \(see page 9\)](#)
- [About Agent Configuration \(see page 11\)](#)
- [About the Agent Launcher and Agent Startup \(see page 13\)](#)
- [About Agent - Server Communication \(see page 15\)](#)
- [Agent Directory Structure \(see page 17\)](#)
- [About Sizing Profiles in vFabric Hyperic \(see page 18\)](#)

Hyperic Security Features and Recommendations

- [SSL Best Practices for New Hyperic Installations \(see page 7\)](#)
- [Agent-Initiated Communication \(see page 7\)](#)
- [Password Encryption \(see page 7\)](#)
- [Protection of Sensitive Data \(see page 8\)](#)
- [Use LDAP Authentication \(see page 8\)](#)

SSL Best Practices for New Hyperic Installations

Hyperic 4.6 and later supports the use of SSL communication for both server-to-agent and agent-to-server communications. VMware recommends that you install trusted keystores from your CA for Hyperic 4.6.x components. If you do not configure your own SSL keystores for Hyperic Agents and the Hyperic Server, the components will generate keystores with self-signed certificates — this is not recommended.

For more information about SSL communications, configuration, and default behavior in Hyperic, see [About SSL in Hyperic \(see page 9\)](#).

Agent-Initiated Communication

You can configure Hyperic such that Hyperic Agents initiate all communications with the Hyperic Server. This feature — referred to as *unidirectional communication* is useful if your network topology requires managed platforms to initiate all communication with processes outside a firewall.

You can configure unidirectional communication interactively at first agent startup, or with the `agent.setup.unidirectional` in the agent's `agent.properties` file. See the requirements described at `agent.setup.unidirectional`.

Password Encryption

This section has information about the credentials that the Hyperic Server and the Hyperic Agent must provide to related product components during normal operation, and how those credentials are supplied, saved, and secured.

Regardless of how the passwords used by the server or agent are supplied, or where they are saved, the password values are always encrypted. Specifically:

- The Hyperic Server supplies a username and password to connect to the Hyperic database. In Hyperic 4.5 and later, the Hyperic installer prompts for the database credentials during the installation process, and saves them in `ServerHome/conf/hq-server.conf`. The value of the password is encrypted.
- The Hyperic Agent supplies a username and password to connect to the Hyperic Server. There are two ways to define the credentials:
 - You can define the username and password for connecting to the server interactively, the first time you start the agent. Upon successful first connection to the server, the agent saves the credentials in its `/data` directory. In Hyperic 4.6.6 and later, the value of the password is encrypted.
 - You can define the username and password for connecting to the server in the agent's `agent.properties` file prior to starting the agent for the first time. At first startup, the agent will use the credentials specified in the properties file, and upon successful connection to the server, the agent will (as with interactive configuration) save the credentials in its `/data` directory, encrypting the password. In Hyperic 4.6.6 and later, the agent will also remove the plain text password from the properties file, and add a password property definition with an encrypted value. Note that Hyperic 4.6.6 and later, you can add encrypted properties to `agent.properties` yourself, as described in [Encrypt Agent Property Value \(see page 99\)](#).
- If the server and agent have user-managed keystores, each component stores the password to its keystore in its properties file.
 - The Hyperic's Server's keystore can be configured interactively at installation. The keystore path and password are saved in `hq-server.conf`, with the password encrypted. If you define the keystore password in the properties file yourself, after installation, In Hyperic 4.6.6 and later, the server will encrypt the password value at next startup.
 - The Hyperic's Agent's keystore can only be configured in the `agent.properties` file. In Hyperic 4.6.6 and later, you can add encrypted properties to `agent.properties` yourself, as described in [Encrypt Agent Property Value \(see page 99\)](#). Otherwise, if you define the password in plain text in the properties file, the agent will encrypt the value at next startup.

Protection of Sensitive Data

In this version of Hyperic, the Hyperic installer writes some sensitive data to installation log files.

Sensitive Data Best Practices

After successfully installing Hyperic Server, delete `InstallerHome/logs/hq-install.log` and `hq-install.log.verbose`, or the whole the exploded installer.

Use LDAP Authentication

Hyperic Server encrypts user passwords using a encryption key you supply during installation. Note however, that Hyperic Server does *not* have a strength-of-password policy, or a lockout policy for failed login attempts. Best practice is to integrate Hyperic with your existing enterprise directory. For information about integrating Hyperic with LDAP, Active Directory, and Kerberos, see [Configure LDAP Properties](#) and [Configure Kerberos Properties](#).

About SSL in Hyperic

- [Hyperic SSL Support](#) (see page 9)
- [Overview of Hyperic 4.6.x SSL Configuration and Defaults](#) (see page 9)
- [Hyperic 4.6.x SSL Certificate Processing](#) (see page 10)
- [SSL Between a 4.6.x Server and 4.5.x Agents](#) (see page 11)
- [SSL and Hyperic Product Plugins](#) (see page 11)

About this page...

This page has information about how SSL communications work in Hyperic. Read this page to understand Hyperic's default behaviors and SSL configuration options and tasks. For instructions on configuring SSL communications and certificates, see:

- *For new Hyperic installations — [Hyperic Installation and Startup Process](#) (see page 20)*
- *For upgrade installations — [Hyperic Upgrade Processes](#) (see page 28)*
- *To change existing SSL configuration — [Configure SSL Options](#) (see page 35)*

Hyperic SSL Support

Hyperic 4.6 and later supports the use of SSL communication for both server-to-agent and agent-to-server communications. Server-to-agent communication is *always* SSL; you cannot configure Hyperic to use plain HTTP for server-to-agent communication. You configure SSL for agent-to-server communication when you configure agent-server communications, either interactively at first agent startup, or in `AgentHome/conf/agent.properties` using the `agent.setup.camSecure` property.

The Hyperic Agent can manage products over SSL, if the product plugin supports it.

Overview of Hyperic 4.6.x SSL Configuration and Defaults

When the Hyperic 4.6.x server and a Hyperic 4.6.x Agent communicate over SSL, each component validates the other's SSL certificate.

Ideally, you should configure Hyperic components to communicate with each other via SSL at installation time. Detailed instructions are provided in [Hyperic Installation and Startup Process](#) (see page 20). The key tasks are:

- Install the Hyperic Server's keystore before installing the server.

- When you run the Hyperic 4.6.x installer in `-full` mode, it offers you the option of configuring the location and password of an existing keystore on the Hyperic Server host. Choose the "user-managed keystore" option and supply the keystore path and password during the server installation dialog.

Default Server Keystore

If you do *not* configure the server to use an existing keystore, and supply its location and password during server installation, the installer will create a keystore for the server with a self-signed certificate. The keystore, named `hyperic.keystore`, will be in `ServerHome/conf`, and have the password "hyperic". The server will present the self-signed certificate when communicating with agents.

- Install the Hyperic Agent's keystore *prior* to first startup — You *must* configure the keystore path and password (and also its alias, if you configure unidirectional agent-server communication) in the `agent.properties` file before starting the agent the first time.
- Configure the Hyperic agent to use SSL when contacting the Hyperic Server — You can do so either interactively at first agent startup, or in `AgentHome/conf/agent.properties` using the `agent.setup.camSecure` property.

Default Agent Keystore

If you do *not* configure the keystore information in `agent.properties` before first startup, the agent will create a keystore with a self-signed certificate. If you choose to run with Hyperic-generated keystores, update the password for each Hyperic-generated keystore; edit `agent.keystore.password` in the `agent.properties` file, and restart the agent.

- Import Managed Product Certs — In Hyperic 4.6, Hyperic plugins that connect to managed products over SSL are updated to support certificate verification. To enable management of such products by a 4.6.x agent, it may be necessary to manually import the target server's certificate into the agent keystore if the server's certificate is not trusted. For more information see [Managed Products and SSL \(see page 36\)](#).

Reconfiguring Certs After Installation and Startup

If you did not configure user-managed keystores for Hyperic 4.6.x at initial installation, and wish to reconfigure the server and agent for user-managed keystores, see [Reconfigure Hyperic for Trusted SSL Certificates \(see page 36\)](#) on [Configure SSL Options \(see page 35\)](#).

Hyperic 4.6.x SSL Certificate Processing

The first time a 4.6.x agent initiates a connection to the Hyperic Server (the first time you start the agent after installation), the server presents its SSL certificate to the agent. If the agent trusts the certificate that the server presented, the agent imports the server's certificate into its own keystore.

How the Agent Decides to Trust the Server's Certificate

The agent trusts a server certificate:

- if that certificate already exists in the agent's keystore, or
- if the certificate has the same CA as the agent's certificate.

By default, if the agent does not trust the certificate presented by the server, the agent issues a warning — you can terminate the configuration process, do the SSL setup. If you wish, you can continue the config process, in which case the agent will import the untrusted certificate presented by the Hyperic Server.

By default, the Hyperic Server and the Hyperic Agent do not import untrusted certificates unless you explicitly respond "yes" to the warning prompt described above. Note, however that both components can be configured to accept untrusted certificates automatically, with no warning. For security reasons, this practice is strongly discouraged. Check the values of `agent.setup.acceptUnverifiedCertificate` (in `AgentHome/conf/agent.properties`) and `accept.unverified.certificates` in `ServerHome/conf/hq-server.conf`.

SSL Between a 4.6.x Server and 4.5.x Agents

To ensure backwards compatibility with pre-4.6 Hyperic Agents, the Hyperic Server upgrade procedure does *not* provide a mechanism for configuring the server to access a user-managed keystore.

When a newly-upgraded Hyperic 4.6.x Server starts up, it generates a self-signed SSL certificate.

When a pre-4.6 Hyperic Agent first connects to a newly-upgraded Hyperic 4.6.x server, its self-signed certificate is imported into the server's keystore.

SSL and Hyperic Product Plugins

In Hyperic 4.6.x, Hyperic plugins that connect to managed products over SSL are updated to support certificate verification. To enable management of such products by a 4.6.x agent, it may be necessary to manually import the target server's certificate into the agent keystore if the server's certificate is not trusted. Affected plugins include:

- vSphere
- RabbitMQ

Import of the managed server's certificate is necessary only if the Hyperic Agent cannot verify the certificate. If the agent's keystore contains a CA cert and the managed server's certificate has been signed by that CA, the agent will be able verify the certificate. Otherwise, you should import the certificate of the signing CA, which is preferable to simply importing the managed server's certificate. If you are not sure of all of the CAs for signed certificates, you might consider importing the certificates in your JRE cacert file, which contains certificates for a variety of common CAs.

About Agent Configuration

- [Agent Startup Configuration Data](#) (see page 12)
- [Optional Agent Configuration Properties](#) (see page 12)
- [Supported Locations for agent.properties](#) (see page 12)
- [How to Change Agent Setup Configuration Properties](#) (see page 12)

About this page...

This page has information about agent configuration data - how it is initially supplied, where it is saved, and how to change it.

For information about where agent configuration fits into the Hyperic deployment process, see:

- [Step 4 - Set Up the Hyperic Agent \(see page 24\)](#) on the [Hyperic Installation and Startup Process \(see page 20\)](#) page — for new Hyperic installations
- [Hyperic Upgrade Processes \(see page 28\)](#) — for upgrade installations

Agent Startup Configuration Data

The configuration choices you must supply for the agent to start up specify where and how it communicates with the Hyperic Server, and vice versa. You can supply these setup values either interactively at first startup, or in the agent's `agent.properties` file before first startup. For a list of setup properties and definitions, see [Communication Properties Reference \(see page 89\)](#) section of the [Set Up Agent in Properties File \(see page 84\)](#) page.

After the agent and server establish successful communication upon first agent startup:

- The Hyperic Agent saves the server's connection data in `AgentHome/data`. The agent creates the `/data` directory upon first successful startup. On subsequent startups, the agent will look at the connection data stored in its `/data` directory to determine where and how to connect to the Hyperic Server.
- The Hyperic Server stores the agent's connection data in the Hyperic database.

Optional Agent Configuration Properties

As described above in [Agent Startup Configuration Data \(see page 12\)](#), at first successful startup, the agent saves the server connection data in its `/data` directory, and upon subsequent startups, looks there for the server connection data. If agent finds the server connection data in its `/data` directory, it does not look at `agent.properties` for the connection data.

Other agent properties that configure optional features and behaviors are persisted only in `agent.properties`. For a complete list of agent properties, see [Agent Properties](#).

Supported Locations for `agent.properties`

`agent.properties` is installed in `AgentHome/conf`.

Note however, that agent first honors an external (from the agent installation) location for the properties file: an `.hq` directory under the home directory of the user under which the agent runs. The agent honors this location for `agent.properties` so that if you upgrade an agent by installing a full agent package (rather than just updating the agent bundle), you will not lose the configuration settings defined in `agent.properties`.

You can also ensure that you do not lose agent configuration settings by backing up `Agent/conf/agent.properties` before installing a full package, and restoring the file after installation.

You can upgrade a 4.x agent by installing an agent bundle only - you can perform this upgrade operation from the Hyperic user interface or using a manual procedure. When you upgrade the agent bundle only, the agent's `/conf` directory is not updated — this method preserves the `agent.properties` file within the agent installation through the upgrade.

How to Change Agent Setup Configuration Properties

You configure an agent's setup properties either interactively at first startup, or in `agent.properties` prior to first startup. The properties that relate to where and how to contact the Hyperic server, are saved in the agent's `/data` directory; those related to how the server can reach the agent are stored in the Hyperic database.

If you need to make changes to those configuration values for an agent at a later time, you can delete the agent's `/data` directory, edit the `agent.setup.*` properties in the `agent.properties` file, and restart the agent. You *must* use this method if you wish to change the agent's listen port — you cannot change port settings without restarting the agent.

If you prefer to change the startup configuration (other than agent listen port) without having to restart the agent, run the following command in a command shell, while the agent is running.

```
AgentHome/hq-agent.sh setup
```

This will allow you to interactively supply new values for the setup properties, as described in [Set Up Agent Interactively](#) (see page 82). This will cause all properties in the properties file to be re-read, and take effect.

About the Agent Launcher and Agent Startup

- [Agent Launcher](#) (see page 13)
- [What Happens When an Agent Starts Up](#) (see page 13)
- [Automatic Restart Behavior](#) (see page 14)

About this page...

This page describes the Hyperic Agent launcher and the agent startup process. For information about:

- *Starting an agent* — See [Start, Stop, and Other Agent Operations](#)
- *SSL certificate verification* — See [About SSL in Hyperic](#) (see page 9)

Agent Launcher

The Hyperic Agent launcher is based on the Java Service Wrapper (JSW), a configurable tool that allows Java applications to be run as an NT service or Unix daemon process. It includes fault correction software to automatically restart crashed or frozen JVMs.

JSW runs as a native executable; it invokes and monitors the Hyperic Agent's JVM, based on configuration information provided to the wrapper at startup. In this way, the wrapper supports restarts of the JVM process without stopping the wrapper process itself. The JSW process acts as a watchdog for the JVM process, periodically pinging it for availability.

For more information about the features of the Java Services Wrapper, and how to configure its behavior, see "<http://wrapper.tanukisoftware.org/doc/english/download.jsp>".

What Happens When an Agent Starts Up

A Hyperic Agent needs to know how to connect to the Hyperic Server, and the Hyperic Server needs to know how to connect to the Hyperic Agent — each component needs the IP address and listen port (and other connection properties) to use to establish a connection with the other. As described in [About Agent Configuration \(see page 11\)](#) you can define the connection properties interactively at first agent startup, or in the agent's `agent.properties` file.

When you start a Hyperic Agent:

1. The agent checks to see if there is a `/data` directory that contains the Hyperic Server's connection properties.
 - At first startup, the `/data` directory will not exist — it is created after the first successful connection between agent and server is established.
 - If the agent finds the connection properties, it tries to connect to the Hyperic Server - see step 5 below.
2. If the agent did not find the `/data` directory, it looks for an `agent.properties` file in a hidden directory named `.hq` in the home directory of the user that runs the agent.
 - This directory will not exist unless you have previously created it. This location for the properties file is supported to ensure configuration data is not lost in the event that the complete agent installation is overwritten in an upgrade.
 - If the agent finds the connection properties, it tries to connect to the Hyperic Server - see step 5 below.
3. If the agent did not find the agent-server connection properties in an `agent.properties` file in the hidden `/.hq` directory, it looks at the `agent.properties` file in its `/conf` directory.
 - If the agent finds the connection properties, it tries to connect to the Hyperic Server - see step 5 below.
4. If the agent did not find connection properties in the `agent.properties` file in its `/conf` directory, it prompts for the properties to be supplied interactively in the command shell.
5. Upon obtaining the agent-server connection properties, the agent attempts to connect to the Hyperic Server.
6. Once communications between agent and server have been successfully established:
 - The agent saves the Hyperic Server connection settings in `AgentHome/data`.
 - The Hyperic Server saves the Hyperic Agent's connection settings in the Hyperic database.

Automatic Restart Behavior

As described above, JSW process periodically pings the agent JVM process.

If the JVM process exits with a non-zero exit code, indicating the JVM is hung or crashed, the JSW will restarts the JVM process. Messages in the Agent and JSW logs indicate that the JSW has restarted the Agent. If the JSW cannot restart the JVM process, it will try again, up to a total of five times. You can configure different actions for the JSW, including the action it takes upon a particular JVM process exit code, and the number of times that it will attempt to restart the process.

About Agent - Server Communication

- [Agent-to-Server Communication \(see page 15\)](#)
- [Server-to-Agent Communication \(see page 15\)](#)
- [Unidirectional Agent-Server Communications \(see page 15\)](#)
- [Unidirectional Communication via a Proxy \(see page 16\)](#)

This section is an overview of the communications between a Hyperic Agent and the Hyperic Server.

Agent-to-Server Communication

Upon startup, a Hyperic Agent initiates a communications channel with the Hyperic Server. The agent continuously batches and sends monitoring results for the platform to the server. Agent-to-server data flows over HTTPS via a byte-encrypted XML protocol called Lather.

The agent sends this data to the server:

- **SSL certificate** — When an agent connects to a server using https, it presents an SSL certificate as part of the handshake. For information about certificate verification, see [Hyperic 4.6.x SSL Certificate Processing \(see page 10\)](#) on [About SSL in Hyperic \(see page 9\)](#).
- **plugin status** — When an agent starts up and loads the plugins in its plugin directory, it sends a plugin status report to the server, including the MD5 checksum of each plugin it loaded. This server uses this information for the server-agent plugin synchronization (SAPS) process, described in [About Plugin Sync at Agent Startup on Plugin Deployment and Management in *vFabric Hyperic Administration*](#).
- **metrics** — You can configure the batch size using the `agent.maxBatchSize` property in `agent.properties`
- **event** — You can configure the batch size using `agent.eventReportBatchSize` in `agent.properties`
- **auto-discovery results** — Auto-discovery results are reported after each auto-discovery scan.
 - By default, the default scan runs every 15 minutes. You can configure the frequency using the `autoinventory.defaultScan.interval.millis` in `agent.properties`
 - By default, the run-time scan runs once a day. You can configure the frequency using the `autoinventory.runtimeScan.interval.millis` in `agent.properties`.

Server-to-Agent Communication

Unless a Hyperic Agent is configured for unidirectional communication, the Hyperic Server initiates communication with a Hyperic Agent to provide metric collection schedules, and relay the commands and data issued or configured by authorized users.

Server-to-agent traffic is always over SSL, and includes:

- **SSL certificate** — When the Hyperic Server connects to an agent using https, it presents an SSL certificate as part of the handshake. For information about certificate verification, see [Hyperic 4.6.x SSL Certificate Processing \(see page 10\)](#) on [About SSL in Hyperic \(see page 9\)](#).
- **metric collection schedules**
- **resource control actions**
- **requests to initiate auto-inventory scans**
- **plugin updates** — If, during the SAPS process, the Hyperic Server detects plugin mismatches between the product plugins on the server and those deployed on an agent, the server syncs the agent's plugin population to resolve the mismatches, and restarts the agent. This behavior is described in [About Plugin Sync at Server Startup on Plugin Deployment and Management in *vFabric Hyperic Administration*](#).

Unidirectional Agent-Server Communications

The default communication between an agent and the server is bi-directional — the Hyperic Agent establishes a connection with the Hyperic Server to send the data described in [Agent-to-Server Communication \(see page 15\)](#), and the Hyperic Server establishes a connection with the Hyperic Agent to send the data described in [Server-to-Agent Communication \(see page 15\)](#).

If your security policies dictate, you can configure the agent to initiate all communications with the Hyperic Server. You can configure unidirectional communications interactively, at first agent startup, or in an agent's `AgentHome/conf/agent.properties` file. If you want to change from bidirectional to unidirectional communications after initial agent configuration, see [Configure Unidirectional Agent - Server Communication \(see page 112\)](#).

If an agent is configured for unidirectional communication, the Hyperic Server does not establish a connection to the agent. Instead, at startup, the agent establishes a persistent connection to the Hyperic Server, and polls the server periodically for server communication.

Unidirectional Communication via a Proxy

If you use unidirectional communications, you can configure the Hyperic Agent to communicate with the Hyperic Server via a proxy server. You must configure this behavior in each agent's `agent.properties` file. For more information, see [Set Up Agent in Properties File \(see page 84\)](#).

Agent Directory Structure

The structure of the Hyperic Agent installation directory is shown below.

```
agent-4.6.n
  bin
  bundles
    agent-4.x.y-nnnn
  conf
  data
  log
  wrapper
```

The root of the directory structure shown above as `agent-4.x.y` is typically referred to in Hyperic documentation as the *AgentHome* directory. Key agent directories include:

- `AgentHome/bin` - contains agent start scripts: `hq-agent.sh` and `hq-agent.bat`
- `AgentHome/conf` - contains `agent.properties`
- `AgentHome/data` - this is the directory where the agent saves the connection properties it uses to contact the Hyperic Server.
- `AgentHome/log` - contains:
 - `agent.log`
 - `agent.startup.log`
 - `wrapper.log`

About Sizing Profiles in vFabric Hyperic

Available only in **vFabric Hyperic**

- [Deployment Size and Installation Sizing Profiles](#) (see page 18)
- [Server Property Values for Sizing Profile](#) (see page 18)
- [How to Select an Installation Sizing Profile](#) (see page 19)

About this page...

This page describes how the Hyperic installer in vFabric Hyperic 4.6.5 and later sets the values of key server properties that affect the system resources allocated to server processing.

Tuning Hyperic HQ

Sizing profiles are a vFabric Hyperic feature. The Hyperic HQ installer does not prompt for a sizing profile, and sets related server properties to the values shown in the the "Small (less than 50 platforms)" column of the table in [Server Property Values for Sizing Profile](#) (see page 18). To tune the Hyperic HQ server, manually edit the properties values. For more information, see [Scaling and Tuning Hyperic Performance](#).

Deployment Size and Installation Sizing Profiles


Starting in vFabric 4.6.5, the Hyperic installer prompts you to select a sizing profile based on the size of your Hyperic deployment. Depending on the volume of platforms you will manage, you choose a "small", "medium", or "large" sizing profile.

Based on the profile you select, the installer sets the values of a variety of Hyperic Server sizing properties in `ServerHome\server.conf` and

`ServerHome\hq-engine\hq-server\webapps\ROOT\WEB-INF\web.xml`. The choice you make — "small", "medium", or "large" — determines the values of server properties that specify the Java options used to start the Hyperic Server, the server's JMS memory limits, the Hyperic database connection pool size, and the number of threads available to the Hyperic Server's internal application server.

Server Property Values for Sizing Profile

The table below lists the Hyperic Server properties whose values are set based by the vFabric Hyperic installer, upon the selected sizing profile, and the values associated with "small", "medium", and "large" profiles.

-  Only the "small" sizing profile is supported if you use the embedded PostgreSQL in vFabric Hyperic.

Property	Small (less than 50 platforms)	Medium (50-250 platfoc)
<code>server.jms.highmemory</code>	350	1400
<code>server.jms.maxmemory</code>	400	1600
<code>server.database-minpoolsize</code>	5	20

<code>server.database-maxpoolsize</code>	100	200
<code>server.java.opts</code>	-Djava.awt.headless=true -XX:MaxPermSize=192m -Xmx512m -Xms512m -XX:+HeapDumpOnOutOfMemory Error -XX:+UseConcMarkSweepGC	-Djava.awt. -XX:MaxPerm -Xmx4g -Xms4g -XX:+HeapD Error -XX:+UseCor
<code>tomcat.maxthreads</code> (new in vFabric Hyperic 4.6.5)	500	2000
<code>tomcat.minsparethreads</code> (new in vFabric Hyperic 4.6.5)	50	100
<code>org.hyperic.lather.maxConns</code> (in ServerHome\hq-engine\hq-server\webapps\ROOT \WEB-INF\web.xml)	475	1900

How to Select an Installation Sizing Profile

The Hyperic installer you to select an sizing profile size when you run it with the `-full` qualifier. If you run the Hyperic with no qualifier, or a qualifier other than `-full`, default property values are set in accordance with the "small" profile.

Sizing Profiles and Upgrade Installations

When you run the Hyperic installer with the `-upgrade` qualifier, the new Hyperic Server properties { `tomcat.maxthreads`) and (`tomcat.minsparethreads`) are added to `server.conf`, with values corresponding to the "small" sizing profile. You can apply a different sizing profile after installation, as described in [Change vFabric Hyperic Server Sizing Profile \(see page 111\)](#).

Hyperic Installation and Startup Process

4

- [Step 1 - Obtain Hyperic Installer \(see page 20\)](#)
- [Step 2 - Set Up Hyperic Database \(see page 20\)](#)
- [Step 3 - Set Up Hyperic Server \(see page 22\)](#)
- [Step 4 - Set Up the Hyperic Agent \(see page 23\)](#)
- [Step 5 - Next Steps for New Installations \(see page 27\)](#)

About this page...

This page lists required and optional steps for a new installation of the Hyperic Server and Hyperic Agent. Steps that are not required for all installations are marked "Optional" or "As Necessary".

For upgrade options and instruction see [Hyperic Upgrade Processes \(see page 28\)](#).

-
- 📌 **Note:** If you obtain vFabric Hyperic as part of vFabric Suite (previously known as "vFabric Platform"), first complete the license activation and installation procedures in *Getting Started with vFabric Suite*. Then follow procedures in this document to set up your environment for vFabric Hyperic and complete any remaining Hyperic-specific installation and configuration tasks.
-

Step 1 – Obtain Hyperic Installer

See [Select and Download an Installer \(see page 31\)](#) for instructions.

Step 2 – Set Up Hyperic Database

If you are installing Hyperic for production use, configure an external Hyperic database *before* installing Hyperic Server. In large environments, set up the Hyperic database on a dedicated platform.

Follow the instructions for the database you use:

- [Set Up MySQL \(see page 37\)](#)
- [Set Up Oracle \(see page 41\)](#)
- [Set Up PostgreSQL \(see page 48\)](#)

For information about Hyperic database requirements and recommendations, see:

- *Hyperic Server Supported Databases*

- *Host Machine Requirements*

Step 3 - Set Up Hyperic Server

1. Before installing the server:

- **Configure user-managed keystore for server** (Optional).
 - If you want the Hyperic server to use a keystore that you manage yourself for SSL communication, rather than Hyperic-generated keystores, set up a JKS format keystore for the Hyperic Server on its host and import the SSL certificate for it. Make a note of the full path to the keystore, and its password — you will supply this information when you run the Hyperic installer (in `-full` mode).

Password Requirement for Hyperic Keystores

The Hyperic Server's keystore password and private key password **must** be the same — otherwise, the Hyperic Server's internal Tomcat-based server will be unable to start. For information about why, see "<http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html>".

| For more information, see [About SSL in Hyperic \(see page 9\)](#)

- **Define Server `HQ_JAVA_HOME`** (As necessary).
 - Hyperic platform-specific server installers include a JRE; platform-independent installers do not. Depending on your environment and the installer you use, you may need to define the location of the JRE to ensure that the server can find the JRE you want it to use.
 - For more information, and instructions, see [Configure JREs for Hyperic Components \(see page 33\)](#).
- **Install Oracle JDBC Driver** (Hyperic HQ with database on Oracle).
 - The Hyperic Server in Hyperic HQ does not include the Oracle JDBC driver. You must obtain and install the driver before installing Hyperic Server. To obtain the driver:
 - a. Go to "http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html"
 - b. In the **JDBC Driver Downloads** section, select the link for your version of Oracle.
 - c. On the drivers download page, select `ojdbc6.jar`. (Hyperic Server 4.6.5 requires a 1.6 JRE).
 - d. On the page that appears accept the licensing agreement.
 - e. Download the jar file.
 - f. Copy the driver jar file to `hyperic-hq-installer/installer-4.x.y/lib`.
- **Make sure the Hyperic database is available.**
 - during the installation process, the Hyperic Server will test the database connection.

2. Install Hyperic Server.

- Follow the instructions for installer you are using:
 - [Run Hyperic Installer \(see page 53\)](#)
 - [Run Windows MSI Installer \(see page 57\)](#)
 - [Install Hyperic Server RPM \(see page 72\)](#)

3. Install License (vFabric Hyperic only).

- Evaluation versions of vFabric Hyperic have a built-in, time-limited evaluation license — you should not need to install a license.
- If you are installing a licensed version of vFabric Hyperic, follow the the instructions on [Install or Configure vFabric Hyperic License \(see page 114\)](#).

4. Start the Hyperic Server.

- Follow the instructions that apply to your environment:
 - If you installed the Hyperic Server from the VMware yum repository to an RHEL VM, the Hyperic Server is configured to start automatically each time the VM starts up. If you installed the Hyperic Server for a downloaded RPM, following these steps to start the Hyperic Server as a daemon:
 - a. Log in to the Hyperic Server host as `root`.
 - b. Open a terminal window and run the `/etc/init.d/hyperic-hq-server` script with the `start` parameter:

```
/etc/init.d/hyperic-hq-server start
```

- On Unix-Like platforms, to start the Hyperic Server as a daemon:
 - a. Open a command shell.
 - b. Change directory to `ServerHome/bin`, and enter:

```
hq-server.sh start
```

- On Windows platforms, to install the Hyperic Server Windows service and start the service:
 - a. Open a terminal window.
 - b. Change directory to `ServerHome\bin` and enter:

```
hq-server.bat install  
hq-server.bat start
```

- The script will display startup information on stdout, then detach and run in the background.

5. Verify that the Hyperic Server started successfully.

- Verify that the Hyperic Server started successfully by pointing your web browser to the Hyperic user interface.
- After the Hyperic Server starts up, which may take a moment or so, display the Hyperic user interface at the following URL, replacing "ServerHostName" with the server's hostname (or "localhost" from the server host). Replace "7080" with the appropriate port number if you configured the server to listen on a different port.

```
http://ServerHostName:7080
```

- The Hyperic Dashboard should appear. Note that there will not be any resources in inventory until you start a Hyperic Agent.
- If the Hyperic Dashboard does not appear, check `server.log` and `bootstrap.log` in `ServerHome/logs/` for information about server startup status.

Sensitive Data Best Practices

In this version of Hyperic, the Hyperic installer writes some sensitive data to installation log files. After successfully installing Hyperic Server, delete `InstallerHome/logs/hq-install.log` and `hq-install.log.verbose`, or the whole the exploded installer.

Step 4 - Set Up the Hyperic Agent

1. Before installing the agent, or if you have already installed the agent, before starting it for the first time:

- **Configure user-managed keystore for Hyperic Agent** (Optional).
 - If you want the Hyperic Agent to use a keystore that you manage yourself for SSL communication, rather than a Hyperic-generated keystore, set up a JKS format keystore for the Hyperic Agent on its host and import the SSL certificate for it. Make a note of the full path to the keystore, and its password — you will configure this data in the agent's `agent.properties` file.



Password Recommendation for Agent Keystore

The agent keystore password and the private key password must be the same.

- **Import certificates for selected managed products to agent keystore** (Optional).
 - In Hyperic 4.6 and later, Hyperic plugins that connect to managed products over SSL are updated to support certificate verification. To enable management of such products by a 4.6.x agent, it may be necessary to manually import the target server's certificate into the agent keystore if the server's certificate is not trusted. Affected plugins include vSphere and Rabbit MQ. For more information, see [About SSL in Hyperic \(see page 9\)](#).
- **Define Agent HQ_JAVA_HOME** (As necessary).
 - Hyperic platform-specific agent installers include a JRE; platform-independent installers do not. Depending on your environment and the installer you use, you may need to define the location of the JRE to ensure that the agent can find the JRE you want it to use. For more information, and instructions, see [Configure JREs for Hyperic Components \(see page 33\)](#).
- **Open firewall port if necessary.**
 - If there is a firewall blocking incoming traffic to a platform where you will install Hyperic Agents, you must open the agent listen port (by default, 2144 for plaintext or 2443 for SSL) so that the agent will accept connections from the Hyperic Server. Note that the firewall built into Windows blocks remote connections by default; in Windows environments you *must* open the agent listen port.

2. Install the agent, following the instructions for the desired installation method:

- [Install an Agent-Only Package \(see page 56\)](#)
- [Run Hyperic Installer \(see page 53\)](#)
- [Run Windows MSI Installer \(see page 57\)](#)
- [Install Hyperic Agent RPM \(see page 79\)](#)

3. Install database drivers for Hyperic HQ .

- In Hyperic HQ, the plugins packaged with the Hyperic Agent for MSSQL, Oracle, Informix, DB2, and Sybase do not include the database vendor's JDBC plugin. (The database plugins in vFabric Hyperic include the JDBC drivers.) After installing Hyperic HQ you must download and install the vendor-provided JDBC drivers for these plugins to work. Install drivers in:

```
AgentHome/bundles/AgentBundle/pdk/lib/jdbc
```


4. **Configure Hyperic Agent in properties file** (As necessary).

- You can define the properties that enable the Hyperic Agent and Hyperic Server to communicate with each other and other agent behaviors in an agent's `agent.properties` file prior to first startup. If you do not, you can supply the values interactively the first time you start the agent (as described in the following step.)
- Note however, that some agent properties *cannot* be supplied interactively. You *must* edit `agent.properties` before first agent startup, as described in [Set Up Agent in Properties File \(see page 84\)](#) if:
 - You want the agent to use an SSL keystore that you manage, rather than a Hyperic-generated keystore.
 - The agent will manage vSphere components.
 - The agent will connect to the Hyperic Server via a proxy server. (Supported in vFabric Hyperic only.)
- Configuring an agent in its properties file, rather than interactively, is also useful particularly when you deploy a lot of agents.

5. Start the Hyperic Agent for the first time.

- a. Make sure that the Hyperic Server is running.
- b. Follow the instructions that apply to your environment:
 - On Unix-Like platforms, to start the Hyperic Agent as a daemon:
 - Open a command shell.
 - Change directory to `AgentHome/bin`, and enter:

```
hq-agent.sh start
```

- On Windows platforms, to set up the Hyperic Agent to run as a service and start the service:
 - Open a terminal window.
 - Change directory to `ServerHome\bin` and enter:

```
hq-agent.bat install
hq-agent.bat start
```

- If you configured all required agent properties in `agent.properties`, the agent will start up and discover the resources on the platform. Otherwise, the agent will prompt for required properties that have not been defined. For instructions on the configuration prompts, see [Set Up Agent Interactively](#) (see page 82).

SSL Warnings?

By default, if the agent does not trust the certificate presented by the server, the agent issues this warning:

```
The authenticity of host 'ServerHost' can't be established. Are you
sure you want to continue connecting? [default=no]:
```

If this warning is issued, for strongest security, press **Return** to terminate the agent configuration process. Before restarting the agent configuration process, obtain an SSL certificate for the agent, and one for the server, both signed by the same trusted CA. Then follow the instructions in [Configure SSL Options](#) (see page 35). If you enter "yes" to continue the agent configuration process, the agent displays the following warning, prompting you to confirm your choice before it imports the certificate that the server presented:

```
The server to agent communication channel is using a self-signed
certificate and could not be verified
Are you sure you want to continue connecting? [default=no]: yes
```

6. Import discovered resources.

- After the Hyperic Agent starts up and runs the auto-discovery process, the resources discovered on the platform appear in the **Auto-Discovery** portlet on the Hyperic Dashboard.
- Open the Hyperic user interface as described above in [View the Hyperic Dashboard \(see page 23\)](#).
- For information about aut-discovery and related instructions, see [Discover and Import Resources to Inventory](#).
- If no resources appear in the **Auto-Discovery** portlet, check the log files (`agent.log`, `agent.log.startup`, and `wrapper.log`) in `AgentHome/logs` to make sure the agent started up successfully.

Step 5 - Next Steps for New Installations

If your Hyperic database runs on Oracle, see:

- [Enable Row Movement \(see page 43\) on Set Up Oracle \(see page 41\)](#)
- [Tuning Hyperic Database on Oracle for Medium to Large Environments \(see page 43\) on Set Up Oracle \(see page 41\)](#)

If your Hyperic database runs on MySQL, see:

- [Tune the Batch Aggregate Inserter for MySQL \(see page 41\) on Set Up MySQL \(see page 37\)](#)

If you are implement a large deployment, see:

- [Scaling and Tuning Hyperic Performance](#)

For information about establishing your monitoring environment see *vFabric Hyperic Administration* for instructions on setting up:

- [Resource groups and applications](#)
- [Users and roles in \(vFabric Hyperic only\)](#)
- [Alerts for resources and resource types](#)
- [Configurable alert notification types](#)
- [Resource auto-discover options](#)

Hyperic Upgrade Processes

- [Hyperic 4.6 Upgrade Options](#) (see page 28)
- [Upgrade Hyperic Server Only](#) (see page 28)
- [Upgrade Server and Agent with Hyperic-Managed Keystores](#) (see page 28)
- [Upgrade Server and Agent with User-Managed Keystores](#) (see page 29)

About this page...

This page has process guidelines for upgrading Hyperic components to version 4.6. Instructions are provided for several upgrade strategies. Consider your deployment and your security requirements in choosing the appropriate upgrade process. See [About SSL in Hyperic](#) (see page 9) for more information.

For information about performing a new installation of Hyperic components, see [Hyperic Installation and Startup Process](#) (see page 20).



Server Agent Plugin Synchronization (SAPS) Requires Server and Agent Upgrade.

In Hyperic 4.6.5, the SAPS process will *not* synchronize plugins on agents of an earlier version. The Hyperic 4.6.5 Server can only synchronize plugins on agents running the same version as the server. For more information about SAPS, see [Plugin Deployment and Management](#).

Hyperic 4.6 Upgrade Options

The upgrade options are:

- Upgrade Hyperic Server only — Hyperic Server 4.6 is backward-compatible with 4.x Hyperic Agents. You can upgrade the Hyperic Server to 4.6, and run a previous (4.x) version of the Hyperic Agent. In this case, the server will use self-signed certificates for SSL communications.
- Upgrade Hyperic Server and Hyperic Agents with Hyperic-Managed Keystores — By default, the upgraded 4.6 Hyperic Server and 4.6 Hyperic Agents will use self-signed certificates for SSL communications.
- Upgrade Hyperic Server and Hyperic Agent with User-Managed Keystores — For best security, you can upgrade the Hyperic Server and all agents to 4.6, and configure the components to use keystores you configure yourself and trusted certificates from a certificate authority.

Upgrade Hyperic Server Only


To upgrade the Hyperic Server only, follow the instructions in [Upgrade Hyperic Server](#) (see page 106).

Upgrade Server and Agent with Hyperic-Managed Keystores

1. Upgrade the Hyperic Server, following the instructions in [Upgrade Hyperic Server \(see page 106\)](#).
2. Upgrade all agents reporting to the Hyperic Server, following the instructions in [Upgrade Hyperic Agent \(see page 103\)](#).

Upgrade Server and Agent with User-Managed Keystores

1. Create a SSL keystore in JKS format with certificates from your CA on the Hyperic Server host, and on each agent host.
2. Upgrade the Hyperic Server, following the instructions in [Upgrade Hyperic Server \(see page 106\)](#).
3. Restart the Hyperic Server.
4. Upgrade all agents reporting to the Hyperic Server to 4.6, following the instructions in [Upgrade Hyperic Agent \(see page 103\)](#).
 - At this point, Hyperic Server and Hyperic Agents are running with self-signed certificates.
5. Shut down all Hyperic Agents.
6. Shut down the Hyperic Server.
7. Configure the Hyperic Server to use the keystore you created on its host. Edit `ServerHome/conf/hq-server.conf`:
 - Define the full path to the server keystore using the `server.keystore.path` property.
 - Define the password for the server keystore using the `server.keystore.password` property.
 - Verify that the `accept.unverified.certificates` is set to "false".
8. Configure each Hyperic Agent to use the keystore you created on its host. For each agent, edit `AgentHome/conf/agent.properties`:
 - Uncomment `agent.keystore.path` and set it to the full path to the keystore you created on the agent host.
 - Uncomment `agent.keystore.password` and set it to the password for the keystore you created on the agent host.

 The first time you start the upgraded agent, the keystore password will be encrypted.

- If the agent is configured for unidirectional communication (`agent.setup.unidirectional=yes`), add `agent.keystore.alias` to the properties file, and set it to the alias for the keystore's primary certificate/private key entry.
- Verify that `agent.setup.acceptUnverifiedCertificate` is "false".
9. Start the Hyperic Server.
10. Start all Hyperic Agents reporting to the Hyperic Server

Step-by-Step Procedures

About this page...

The following topics have instructions for performing Hyperic installation and upgrade tasks. To see where each task fits into the implementation process see:

- [Hyperic Installation and Startup Process \(see page 20\)](#) — for new installations
 - [Hyperic Upgrade Processes \(see page 28\)](#) — for upgrade installations
-
- [Select and Download an Installer \(see page 31\)](#)
 - [Configure JREs for Hyperic Components \(see page 33\)](#)
 - [Configure SSL Options \(see page 35\)](#)
 - [Set Up MySQL \(see page 37\)](#)
 - [Set Up Oracle \(see page 41\)](#)
 - [Set Up PostgreSQL \(see page 48\)](#)
 - [Run Hyperic Installer \(see page 53\)](#)
 - [Install an Agent-Only Package \(see page 56\)](#)
 - [Run Windows MSI Installer \(see page 57\)](#)
 - [Install Hyperic Server RPM \(see page 72\)](#)
 - [Install Hyperic Agent RPM \(see page 79\)](#)
 - [Set Up Agent Interactively \(see page 82\)](#)
 - [Set Up Agent in Properties File \(see page 84\)](#)
 - [Encrypt Agent Property Value \(see page 99\)](#)
 - [Install Hyperic Agents in Volume \(see page 100\)](#)
 - [Upgrade Hyperic Agent \(see page 103\)](#)
 - [Upgrade Hyperic Server \(see page 106\)](#)
 - [Uninstalling an Agent \(see page 110\)](#)
 - [Change vFabric Hyperic Server Sizing Profile \(see page 111\)](#)
 - [Configure Unidirectional Agent - Server Communication \(see page 112\)](#)
 - [Install or Configure vFabric Hyperic License \(see page 114\)](#)

Select and Download an Installer

- [Key Facts About Hyperic Installer Packaging](#) (see page 31)
- [Installer Types](#) (see page 31)
- [Hyperic Installer Features](#) (see page 32)
- [Where to Get Hyperic Installation Packages](#) (see page 33)

About this page...

This page has instructions for selecting the right Hyperic installation package. After obtaining the appropriate installation package, see:

- [Hyperic Installation and Startup Process](#) (see page 20) — *For instructions on doing a new installation.*
- [Hyperic Upgrade Processes](#) (see page 28) — *For instructions on doing an upgrade installation.*

Key Facts About Hyperic Installer Packaging

- About JRE packaging
 - Most Hyperic installation packages include 1.6 JREs, for both the Hyperic Agent and the Hyperic Server.
 - Hyperic installers referred to as *platform-independent* do not include a JRE — use a platform-independent package if you have a pre-existing JRE on the target platform that you want the server or agent to use. A platform-independent package for Unix-like environments is also appropriate if there is no Hyperic installer for your operating system.
 - The Agent RPM does not include a JRE.
- About Hyperic database options
 - Platform-independent server packages do not support the option of installing Hyperic Server with a built-in database.
 - The MSI installer automatically installs a built-in PostgreSQL database for Hyperic Server; you cannot install Hyperic Server to use an external database with the MSI installer.

Installer Types

There are several ways you can install Hyperic components:

- **Hyperic Installer** — When you are installing the Hyperic Server for a production environment, or want to explicitly configure key behaviors, this is the type of installer to use. The installer is script-based and flexible — you can do a quick install that sets up defaults for virtually all Hyperic Server configuration options, or run it in "full" mode to respond to the configuration dialog yourself. You can also use this installer to install the Hyperic Agent. To install an agent, the installer simply creates a home directory for the agent, and unpacks the agent archive into it — you configure agent behavior by editing its properties file prior to first startup, or via an interactive dialog at first startup. Full installer packages are available for both Windows and Unix-like environments, with and without built-in JREs. See rows 1 and 2 in [Hyperic Installer Features](#) (see page 32) below.
- **Agent Archives** — Agent-only archives are useful when you roll out agents to a large number of platforms with varying operating systems and architectures. Agent archives are available for both Windows and Unix-like environments, with and without built-in JREs. See rows 3 and 4 in [Hyperic Installer Features](#) (see page 32) below.
- **RHEL RPM** — RPMs are available. Note that the Hyperic RPM is the standard Hyperic installer wrapped in an Expect script. See rows 5 and 6 in [Hyperic Installer Features](#) (see page 32) below.

- **InstallShield MSI** — The MSI installer for Windows environments is most useful if you are a new evaluator of Hyperic, and prefer a GUI interaction, rather than a command-line dialog. The MSI is not suitable for production environments, as it does not provide the option of installing Hyperic Server to work with an external database. The MSI installs a built-in PostgreSQL database, the use of which is unsuitable for production use. See rows 7 in [Hyperic Installer Features \(see page 32\)](#) below.

Hyperic Installer Features

Installer Package Type	Architectures
1 Hyperic Full Installer (tar)	Linux (32 bit) Linux (64 bit) Solaris Sparc (32 or 64 bit) Mac OSX (32 or 64 bit) Platform-independent (32 or 64 bit)
2 Hyperic Full Installer (zip)	Windows (32 bit) Platform-independent (32 or 64 bit)
3 Agent archive (tar)	Linux (32 bit) Linux (64 bit) Solaris Sparc (32 or 64 bit) Mac OSX (32 or 64 bit) AIX (32 or 64 bit) HP-UX 11.x (32 or 64 bit) Platform-independent (32 or 64 bit)
4 Agent archive (zip)	Windows (32 bit) Platform-independent (32 or 64 bit)
5 Server RPM	RHEL x86-64 (64 bit) RHEL i386 (32 bit) Platform-independent (32 or 64 bit)
6 Agent RPM	Platform-independent (32 or 64 bit)
7 MSI	Windows (32 bit)

No 64-bit Installers for Windows Environments

Hyperic installers for Windows include 32-bit JREs — there is no Hyperic package for Windows with a 64-bit JRE. You can install Hyperic component from the Hyperic 32-bit package for Windows, and configure the component to use a pre-existing 64-bit JRE on the platform. See [Configure JREs for Hyperic Components \(see page 33\)](#).

Known Hyperic Agent Problem on Windows 2008 R2 x64

A known problem ("<https://jira.hyperic.com/browse/HHQ-4640>" with the Hyperic Agent 4.6, installing the agent on Windows 2008 x64 (Microsoft Windows Version 6.1.7600) causes the agent's Java Service Wrapper to crash and issue messages similar to:

```
C:\Program Files
(x86)\Hyperic\4.6.0-EE-110\hyperic-hqee-agent-4.6.0.BUILD-SNAPSHOT\bin>hq-agent.bat install
Out of memory in logging code (PPB1)
Out of memory in logging code (PPB1)
Logging mutex wait failed.
Logging mutex wait failed.
...
```

The agent service is successfully installed, despite the errors.

Note for Solaris x86 Environment

There is *no* platform-specific Hyperic installer for Solaris x86. Install a no-JRE Hyperic package in Solaris x86 environments and configure the components to use pre-existing JREs in your environment. For more information, see [Configure JREs for Hyperic Components](#) (see page 33).

Where to Get Hyperic Installation Packages

vFabric Hyperic and Hyperic HQ installers are available on the VMware download page at "<http://downloads.vmware.com/>". Navigate to the page and choose **VMware vFabric Hyperic**.

vFabric Hyperic RPM packages are available in the VMware RPM repository. For information about installing from the repository, see "<http://pubs.vmware.com:8080/vfabric5/topic/com.vmware.vfabric.platform.5.0/vfabric/install-repo-install.html>".

If you plan a local RPM installation, rather than from the repository, vFabric Hyperic RPMs are available for download at "<http://repo.vmware.com/pub/rhel5/vfabric/5/i386>" and "http://repo.vmware.com/pub/rhel5/vfabric/5/x86_64".

Configure JREs for Hyperic Components

- [About Hyperic JREs](#) (see page 33)
- [Hyperic Agent JRE on Unix](#) (see page 34)
- [Hyperic Agent JRE on Windows](#) (see page 34)

About this page...

This page has information about how to configure Hyperic components to use a particular JRE.

It is not generally necessary to configure the location of the JREs used by the Hyperic Server or Hyperic Agents. For more information, see [About Hyperic JREs](#) (see page 33) below.

If you do need to configure the location of the JRE for a Hyperic component, define the environment or system variable, as described below, and restart the system before starting the Hyperic component.

About Hyperic JREs

Both Hyperic Server and Hyperic Agents require a JRE. Some Hyperic installers include a JRE; they are called *platform-specific*. Installers without a JRE are called *platform-independent*. Depending on your environment and the installation package you use, you may need to define the location of the JRE for the server or your agents.

- You *do not* need to configure the location of the JRE when you do a platform-specific server install on a machine with no JRE.
- You *do* need to configure the location of the JRE when you:
 - Install the Hyperic Server (or the Hyperic Agent) with a platform-independent installer,
 - Install the Hyperic Server (or the Hyperic Agent) with a platform-specific installer on a machine that has its own JRE that you prefer to use.
 - Install a Hyperic Agent from an RPM. Agent RPMs do not include a JRE.

How Hyperic Server Resolves its JRE on any Platform

On both Unix-Like and Windows platforms, Hyperic Server resolves the JRE to use in this order:

1. HQ_JAVA_HOME environment variable
2. embedded JRE
3. JAVA_HOME environment variable

The Hyperic Agent has resolves its JRE differently than the Hyperic Server does, and differently depending on platform type. For more information, see the appropriate section:

- [Hyperic Agent JRE on Unix \(see page 34\)](#)
- [Hyperic Agent JRE on Windows \(see page 34\)](#)

Hyperic Agent JRE on Unix

The Hyperic Agent installer, like the server installer, is available with or without a JRE. If you do a platform-specific agent install on a machine without an existing JRE, you do not need to explicitly configure the location of the JRE to use.

You should configure the JRE location with the HQ_JAVA_HOME environment variable if you do a platform-independent agent install, or a platform-specific install on a Unix machine that already has a JRE that you prefer to use.

How a Hyperic Agent Resolves its JRE on Unix

On Unix-like platforms, Hyperic Agent resolves the JRE to use in this order:

1. HQ_JAVA_HOME environment variable
2. Embedded JRE
3. JAVA_HOME

Hyperic Agent JRE on Windows

The Hyperic Agent installer, like the server installer, is available with or without a JRE.

You should configure the JRE location with the `HQ_JAVA_HOME` system variable if you do a platform-independent agent install on Windows, or a platform-specific install on a Windows machine that already has a JRE that you prefer to use.

 **How a Hyperic Agent Resolves its JRE on Windows**

On Windows platforms, Hyperic Agent resolves the JRE to use in this order:

1. `HQ_JAVA_HOME` **system** variable (*not* an environment variable)
2. Embedded JRE

To define a system variable, use:

My Computer > Properties > Advanced > Environment Variables > System variables > New

Configure SSL Options

- [SSL Setup for New Hyperic Installations \(see page 35\)](#)
- [SSL Setup for Upgrade Installations \(see page 36\)](#)
- [Managed Products and SSL \(see page 36\)](#)
- [Reconfigure Hyperic for Trusted SSL Certificates \(see page 36\)](#)
- [Reconfigure Hyperic for Self-Signed Certificates \(see page 37\)](#)

About this page...

This page has information about configuring Hyperic components for user-managed keystores. If you do not configure the Hyperic Server and Hyperic Agents to use keystores you establish and manage, they will generate default keystores with self-signed certificates.

Hyperic recommends user-managed keystores. For more information, see [About SSL in Hyperic \(see page 9\)](#).


The information on this pages relates to the [Configure User-Managed Keystore for Server \(see page 22\)](#) and [Configure User-Managed Keystore for Agent \(see page 24\)](#) steps in the [Hyperic Installation and Startup Process \(see page 20\)](#).

SSL Setup for New Hyperic Installations

This section summarizes the key steps in configuring a new Hyperic 4.6 deployment for user-managed keystores. To see how these steps fit into the overall installation and startup process, see [Hyperic Installation and Startup Process \(see page 20\)](#).

1. Obtain SSL certificates for the Hyperic Server and each Hyperic Agent.
2. Set up a JKS format keystore for the Hyperic Server on its host, import the SSL certificate for it, and note the full path to the keystore and its password. When you run the Hyperic installer in `-full` mode, the installer prompts for this information.

3. Setup a keystore for each Hyperic Agent on its host, import the SSL certificate for it, and configure its location and password in the agent's `AgentHome/conf/agent.properties` file, by setting the values of `agent.keystore.path` and `agent.keystore.password`.

 Password Requirement for Hyperic Keystores

The Hyperic Server's keystore password and private key password **must** be the same — otherwise, the Hyperic Server's internal Tomcat-based server will be unable to start. For information about why, see "<http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html>". Follow the same convention for a Hyperic Agent keystore — set the password for the agent keystore be the same as the agent private key,

4. If you plan to configure Hyperic Agents for unidirectional communication, define the keystore name using the `agent.keystore.alias` property.
5. Restart each agent after editing its properties file.

SSL Setup for Upgrade Installations

Please see [Hyperic Upgrade Processes \(see page 28\)](#).

Managed Products and SSL

In Hyperic 4.6, Hyperic plugins that connect to managed products over SSL are updated to support certificate verification. To enable management of such products by a 4.6 agent, it may be necessary to manually import the target server's certificate into the agent keystore if the server's certificate is not trusted. Affected plugins include:

- vSphere
- RabbitMQ

Import of the managed server's certificate is necessary only if the Hyperic Agent cannot verify the certificate. If the agent's keystore contains a CA cert and the managed server's certificate has been signed by that CA, the agent will be able verify the certificate. Otherwise, you should import the certificate of the signing CA, which is preferable to simply importing the managed server's certificate. If you are not sure of all of the CAs for signed certificates, you might consider importing the certificates in your JRE cacert file, which contains certificates for a variety of common CAs.

Reconfigure Hyperic for Trusted SSL Certificates

This section has instructions for changing Hyperic's SSL certificate configuration from default, Hyperic-generated keystores to user-managed keystores.

1. Install and configure a trusted PKCS12 format keystore for Hyperic Server:
 - a. Obtain an SSL certificate from your CA and install it on the Hyperic Server host.
 - If your certificate is not PKCS12 format, you can use the **openssl** tool to convert it. For more information, see "<http://community.jboss.org/wiki/sslsetup>".
 - After ensuring your certificate is in the correct format, use **java-keytool** to install it. For more information, see "<http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/keytool.html>".
 - b. Open `ServerHome/conf/hq-server.conf` in a text editor.
 - c. Set the value of `accept.unverified.certificates` to "false".
 - d. Define the location of your trusted keystore with the `server.keystore.path` property.
 - e. Define the password for your trusted keystore with the `server.keystore.password` property.
 - f. Save your changes.
 - g. Restart the Hyperic Server.
2. For each Hyperic 4.6 Agent reporting to the Hyperic Server:
 - a. Obtain an SSL certificate from your CA and install it on the Hyperic Agent host.
 - b. Open `AgentBundle/AgentHome/agent.properties` in a text editor.
 - c. Set the value of `agent.setup.acceptUnverifiedCertificate` to "false".
 - d. Define the location of your trusted keystore with the `agent.keystore.path` property.
 - e. Define the password for your trusted keystore with the `agent.keystore.password` property.
 - f. Save your changes.
 - g. Restart the Hyperic Agent.

Reconfigure Hyperic for Self-Signed Certificates

This section has instructions for changing Hyperic's SSL certificate configuration from user-managed keystores to default, Hyperic-generated keystores.

Default Certs Not Recommended

For best security, do not configure Hyperic to use self-signed certificates.

1. Open `ServerHome/conf/hq-server.conf` in a text editor.
 - a. Set the value of `accept.unverified.certificates` to "true".
 - b. Restart the Hyperic Server.
2. For each Hyperic 4.6 Agent reporting to the Hyperic Server:
 - a. Open `AgentBundle/AgentHome/agent.properties` in a text editor.
 - b. Set the value of `agent.setup.acceptUnverifiedCertificate` to "true".
 - c. Save your changes.
 - d. Restart the Hyperic Agent.

Set Up MySQL

*Topics marked with * relate to features available only in vFabric Hyperic.*

- [Set Up Hyperic Database on MySQL \(see page 38\)](#)
 - [Step 1 - Create a MySQL Database Instance \(see page 38\)](#)
 - [Step 2 - Configure MySQL Startup Options and System Variables \(see page 38\)](#)
 - [Step 3 - Verify Database Setup \(see page 40\)](#)
- [Solve Problems with MySQL Startup \(see page 40\)](#)
- [Tune the Batch Aggregate Inserter for MySQL \(see page 41\)](#)

 For information about maintaining a MySQL Hyperic database, see MySQL Maintenance Examples.

Set Up Hyperic Database on MySQL

This section has instructions for setting up MySQL as your external Hyperic database.

This task corresponds to [Step 2 – Set Up Hyperic Database](#) (see page 20) of [Hyperic Installation and Startup Process](#) (see page 20).

It is assumed that you have already installed MySQL and are either familiar with MySQL or have the support of someone who is.

Note: If you are installing Hyperic for evaluation, you can use Hyperic's built-in PostgreSQL database, rather than set up an external database.

Step 1 - Create a MySQL Database Instance

Run these commands at the mysql prompt, as the root user:

```
mysql> create user 'hqadmin'@'<hq_server_host>' identified by '<passwd>';
mysql> create database HQ CHARACTER SET utf8 COLLATE utf8_bin;
mysql> grant all on HQ.* to 'hqadmin'@'<hq_server_host>';
```

UTF8 is required for encoding.

Step 2 - Configure MySQL Startup Options and System Variables

In this step, you configure the MySQL database by editing the settings in its configuration file. For Unix-like environments, the file is `/etc/my.cnf`. On Windows the file is `my.ini`, located in the MySQL installation base directory.

For more information about InnoDB startup options and system variables, see "<http://dev.mysql.com/doc/refman/5.0/en/innodb-parameters.html>".

1. Enable the full query log. Every query (even ones with incorrect syntax) that the database server receives will be logged. This is useful for debugging, but it is usually disabled in production use. Be sure to change the paths given here to match your environment.

```
[mysqld]
log-error = mysqld.err
log = mysql_general.log
```

2. Print warnings to the error log file. If you have any problem with MySQL, you should enable logging of warnings and examine the error log for possible explanations.

```
log_warnings
```

3. Configure buffer pool size. The size of the MySQL buffer pool is has a significant impact on MySQL performance. If your database is on a dedicated machine, make the buffer pool about 80% of total memory.

```
innodb_buffer_pool_size = 256M
```

4. Configure the frequency with which the log buffer is written to the log, and the point at which the log is flushed to the disk. Setting `innodb_flush_log_at_trx_commit` to 0 dramatically increases MySQL performance, but with this setting, you are likely to lose data in the event of a server crash. If loss of data is unacceptable, set `innodb_flush_log_at_trx_commit` to 2. Hyperic does not recommend setting the value to 1.

```
innodb_flush_log_at_trx_commit = 2
innodb_log_buffer_size = 64M
innodb_log_file_size = 256M
```

5. Configure `innodb` as the default storage engine. (Required.)

```
default-storage-engine=innodb
bulk_insert_buffer_size = 32M
join_buffer_size = 8M
max_heap_table_size = 256M
tmp_table_size = 256M
max_tmp_tables = 48
myisam_sort_buffer_size = 256M
```

6. Configure the sort buffer size. MySQL recommends a `sort_buffer_size` larger than the one suggested her.

```
sort_buffer_size = 64K
```



An article on experimenting with sort buffer size is available at "<http://www.mysqlperformanceblog.com/2007/08/18/how-fast-can-you-sort-data-with-mysql>".

7. Configure the read buffer size. Because Hyperic does a significant volume of sequential reads, a large read buffer improves performance.


```
read_buffer_size = 1M
read_rnd_buffer_size = 10M
table_cache = 2048
set-variable = max_connections=400
key_buffer_size = 256M
thread_cache_size = 32
```

8. Configure the number of threads that can run in the InnoDB kernel. A starting point for setting this value is to set a value equal to 2 times the number of CPUs times the number of disks.

```
innodb_thread_concurrency = 8
```

9. Set the method that is used to flush data and log files. For battery-backed-up storage with write-back cache mode on Linux systems, the `O_DIRECT` flush method is good. For information on other innodb flush methods see "http://dev.mysql.com/doc/refman/5.0/en/innodb-parameters.html#option_mysql_innodb_flush_method".

```
innodb_flush_method=O_DIRECT
innodb_rollback_on_timeout=1
```

-
-  On Windows the flush method is always `async_unbuffered`. You do not need to set `innodb_flush_method=O_DIRECT` on Windows platforms.
-

- In this situation, tune your Linux OS (version 2.6 or higher) to favor the use of main memory rather than file caches with either:

```
# sysctl -w vm.swappiness=30
```

- or

```
# echo 30 >/proc/sys/vm/swappiness
```

10. Set query cache size. Generally, the higher this value, the better the performance. However, in MySQL versions older than 5.0.50, beware of setting this variable too high, as it may cause the database to pause. For more information, see the bug description at "<http://bugs.mysql.com/bug.php?id=21074>".

```
query_cache_size = 0
```

11. Set query cache limit. The default value here is 1M. If the `qcache_hits-to-qcache_inserts` ratio is low, raise this value.

```
query_cache_limit = 8M
```

12. Set character encoding. Hyperic requires a char encoding of utf-8.

```
default-character-set=utf8
collation_server=utf8_bin
character_set_system=utf8
```

Step 3 - Verify Database Setup

instructions tbd

-
-  After the database is created, you can set up Hyperic Server following the instructions in [Step 3 - Set Up Hyperic Server](#) (see page 22) of the [Hyperic Installation and Startup Process](#) (see page 20).
-

Solve Problems with MySQL Startup


If MySQL fails to start and issues a message similar to this:

```
InnoDB: Error: log file ./ib_logfile0 is of different size 0 5242880 bytes
InnoDB: than specified in the .cnf file 0 268435456 bytes!
080403  8:06:13 ERROR Default storage engine (InnoDB) is not available
080403  8:06:13 ERROR Aborting
```

the actual log size does not match the configured log size.

Delete the log files in `/var/lib/mysql/` and restart MySQL.

Tune the Batch Aggregate Inserter for MySQL

 Do this tuning *after* installing the Hyperic Server.

These tuning recommendations are based on a performance tuning exercise in an environment with 700 Hyperic Agents reporting to an Hyperic Server on an 8 way / 16 GB host with an MySQL database running on an 8 way / 8 GB host, each running CentOS 5, with

- Workers: 4
- QueueSize: 4000000
- BatchSize: 2000

With 7 hours of backfilled data the server peaked out at 2.2 million rows inserted.

This intent of the strategy was to keep the Batch Aggregate Inserter (BAI) on "cruise control", instead throwing threads at the queued metrics all at once and causing CPU spikes.

It was found that the BAI workers had no trouble keeping up with the "normal" incoming load, and in a catchup scenario (after backfilling) the high Queue Size allowed them plenty of time to catch up.

For a smaller deployment, consider only tweaking the number of workers down to 1 or 2. This will ease random CPU spikes and MySQL should have no problem keeping up with the incoming traffic.

Please NOTE these settings may not be applicable to PostgreSQL and Oracle since MySQL handles catchup scenarios much more gracefully.

To update the Batch Aggregate Inserter settings for MySQL run these commands at the `mysql` prompt as the `hqadmin` user:

```
mysql> update HQ.EAM_CONFIG_PROPS set propvalue = 4 where propkey = 'BATCH_AGGREGATE_WORKERS';
mysql> update HQ.EAM_CONFIG_PROPS set propvalue = 2000 where propkey = 'BATCH_AGGREGATE_BATCHSIZE';
mysql> update HQ.EAM_CONFIG_PROPS set propvalue = 4000000 where propkey = 'BATCH_AGGREGATE_QUEUE';
```

Set Up Oracle

Topics marked with * relate to features available only in vFabric Hyperic.

- [Set Up Hyperic Database on Oracle \(see page 42\)](#)
 - [Create Hyperic Database \(see page 42\)](#)
 - [Create Tablespaces \(see page 42\)](#)
 - [Create the Database User \(see page 42\)](#)
 - [Grant Permissions to the Database User \(see page 43\)](#)
- [Set Up Hyperic Server \(see page 43\)](#)
- [Enable Row Movement \(see page 43\)](#)
- [Tuning Hyperic Database on Oracle for Medium to Large Environments \(see page 43\)](#)
 - [Create the TS_HQDB_16K Tablespace \(see page 43\)](#)
 - [Configure REDO Logs \(see page 44\)](#)
 - [Configure Initialization Parameters for Oracle \(see page 44\)](#)
 - [Configure Batch Aggregate Inserter \(see page 44\)](#)
 - [Move Database Tables \(see page 44\)](#)
 - [Rebuild Indexes for Moved Tables \(see page 45\)](#)
 - [Configure Tables for High Concurrency \(see page 45\)](#)
 - [Restart Hyperic Server \(see page 46\)](#)
- [Periodic Oracle Database Maintenance \(see page 46\)](#)

Set Up Hyperic Database on Oracle

This section has instructions for setting up Oracle as your external Hyperic database. It is assumed that you have already installed Oracle and are either familiar with Oracle or have the support of someone who is.


Create Hyperic Database

The database can be created with Oracle Database Configuration Assistant.

Select **New Database** (Includes datafiles = No). To save space, decline to install the Example Schemas.

Assuming Oracle runs on a dedicated host, you can select the "Typical Memory" configuration.

Select "OLTP" as the type of database sizing to use. Allocate as high a percentage of system resources as you can afford: 70-90%, ideally in the higher range. See [Configure Initialization Parameters for Oracle \(see page 44\)](#) for SGA and PGA size.

 Hyperic requires utf-8 character encoding

Create the database with utf-8 character encoding.

Create Tablespaces

1. Create the TEMP_HQDB temporary tablespace, 2 GB in size
2. Create the TS_HQDB tablespace, 25 GB in size. This tablespace will be used to store HQ_METRIC_DATA_*D_*S tables.

Create the Database User

In this step you create the database user account that the server will use to access the Oracle database.

There are multiple methods for creating a user in Oracle. To do it using SQL*Plus, log into the Oracle instance as the system user with SQL*Plus, and issue the create user command:

```
SQL> CREATE USER HQUUSER IDENTIFIED BY HQPASSWORD DEFAULT TABLESPACE TS_HQDB;
```

replacing HQUUSER and HQPASSWORD with desired values.

Grant Permissions to the Database User

You can grant the database user the necessary permissions in SQL*Plus with the grant command:

```
SQL> GRANT CONNECT, RESOURCE, CREATE VIEW TO HQUUSER;
```

Verify the permission setting:

```
SQL> SELECT GRANTED_ROLE, DEFAULT_ROLE FROM dba_role_privs WHERE grantee = 'HQUUSER';
```

Make sure that you see the following rows for CONNECT and{{ RESOURCE}} roles:

```
GRANTED_ROLE DEFAULT_ROLE
CONNECT YES
RESOURCE YES
```

If that is not the case, update the permissions:

```
ALTER USER HQUUSER DEFAULT ROLE RESOURCE, CONNECT;
```

Set Up Hyperic Server

For instructions, see [Step 3 - Set Up Hyperic Server \(see page 22\)](#) of the [Hyperic Installation and Startup Process \(see page 20\)](#).

Enable Row Movement

After installing Hyperic Server, run the `enable_row_movement.sql` procedure attached to this page — this enables the routine maintenance described in [Periodic Oracle Database Maintenance \(see page 46\)](#). It is only necessary to enable row movement once; you do not need to do it on a recurring basis.

Tuning Hyperic Database on Oracle for Medium to Large Environments

If you manage more than 100 platforms, follow the steps in this section to tune your Oracle-hosted Hyperic database.

Create the TS_HQDB_16K Tablespace

Create the `TS_HQDB_16K` tablespace, 25 GB in size, with 16 K blocksize. Here is an example of the syntax to create a table space with 16 K blocksize, where `'datafile_name.dbf'` is a file where the tablespace data will physically reside:

```

CREATE TABLESPACE TS_HQDB_16K
datafile 'datafile_name.dbf' SIZE 25000M AUTOEXTEND OFF
ONLINE
PERMANENT
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
BLOCKSIZE 16K
SEGMENT SPACE MANAGEMENT AUTO
;

```

Configure REDO Logs

Redo logs are transaction journals. Each transaction is recorded in the redo logs. Redo logs are used in a serial fashion with each transaction queuing up in the redo log buffers and being written one at a time into the redo logs.

Configure at least three REDO logs, each 2048 MB in size.

Locate REDO logs on a separate disk spindle from datafiles.

Do not locate REDO logs on a RAID array.

Configure Initialization Parameters for Oracle

Set following Oracle initialization parameters:

```

DB_WRITER_PROCESS = 4
SGA_MAX = 8G
SGA_TARGET = 7G
SHARED_POOL_SIZE=700M
PGA_AGGREGATE_TARGET=1500M
DB_16K_CACHE_SIZE=1000M
DB_KEEP_CACHE_SIZE = 500M
FILESYSTEMIO_OPTIONS=SetAll
DB_FILE_MULTIBLOCK_READ_COUNT=16
OPEN_CURSORS=300
PROCESSES=500

```

Configure Batch Aggregate Inserter

1. Make sure that the Hyperic Server is shut down.
2. Log into the Oracle instance as the Hyperic user with SQL*Plus, and issue these commands to increase the data aggregate inserter batch size to 8000 and number of workers to 10:

```

UPDATE EAM_CONFIG_PROPS SET PROPVALUE=8000 where PROPKEY='BATCH_AGGREGATE_BATCHSIZE';
UPDATE EAM_CONFIG_PROPS SET PROPVALUE=10 where PROPKEY='BATCH_AGGREGATE_WORKERS';
UPDATE EAM_CONFIG_PROPS SET PROPVALUE=5000000 where
PROPKEY='BATCH_AGGREGATE_QUEUE';

```

Move Database Tables

While still logged into the Oracle instance as the system user, run these commands to move metric and measurement tables to the TS_HQDB_16K tablespace:

```

alter table HQADMIN.HQ_METRIC_DATA_0D_0S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_0D_1S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_1D_0S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_1D_1S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_2D_0S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_2D_1S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_3D_0S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_3D_1S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_4D_0S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_4D_1S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_5D_0S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_5D_1S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_6D_0S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_6D_1S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_7D_0S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_7D_1S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_8D_0S move tablespace TS_HQDB_16K;
alter table HQADMIN.HQ_METRIC_DATA_8D_1S move tablespace TS_HQDB_16K;
alter table HQADMIN.EAM_MEASUREMENT_DATA_1D move tablespace TS_HQDB_16K;
alter table HQADMIN.EAM_MEASUREMENT_DATA_1H move tablespace TS_HQDB_16K;
alter table HQADMIN.EAM_MEASUREMENT_DATA_6H move tablespace TS_HQDB_16K;

```

Rebuild Indexes for Moved Tables

Run these commands to rebuild the indexes for the moved tables:

```

alter index HQADMIN.METRIC_DATA_0D_0S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_0D_1S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_1D_0S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_1D_1S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_2D_0S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_2D_1S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_3D_0S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_3D_1S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_4D_0S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_4D_1S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_5D_0S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_5D_1S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_6D_0S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_6D_1S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_7D_0S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_7D_1S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_8D_0S_MID_IDX rebuild;
alter index HQADMIN.METRIC_DATA_8D_1S_MID_IDX rebuild;
alter index HQADMIN.MEASUREMENT_DATA_1H_MID_IDX rebuild;
alter index HQADMIN.MEASUREMENT_DATA_6H_MID_IDX rebuild;
alter index HQADMIN.MEASUREMENT_DATA_1D_MID_IDX rebuild;

```

Configure Tables for High Concurrency

```
alter table <schema>.HQ_METRIC_DATA_0D_0S initrans 15;
alter index <schema>.METRIC_DATA_0D_0S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_0D_1S initrans 15;
alter index <schema>.METRIC_DATA_0D_1S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_1D_0S initrans 15;
alter index <schema>.METRIC_DATA_1D_0S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_1D_1S initrans 15;
alter index <schema>.METRIC_DATA_1D_1S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_2D_0S initrans 15;
alter index <schema>.METRIC_DATA_2D_0S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_2D_1S initrans 15;
alter index <schema>.METRIC_DATA_2D_1S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_3D_0S initrans 15;
alter index <schema>.METRIC_DATA_3D_0S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_3D_1S initrans 15;
alter index <schema>.METRIC_DATA_3D_1S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_4D_0S initrans 15;
alter index <schema>.METRIC_DATA_4D_0S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_4D_1S initrans 15;
alter index <schema>.METRIC_DATA_4D_1S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_5D_0S initrans 15;
alter index <schema>.METRIC_DATA_5D_0S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_5D_1S initrans 15;
alter index <schema>.METRIC_DATA_5D_1S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_6D_0S initrans 15;
alter index <schema>.METRIC_DATA_6D_0S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_6D_1S initrans 15;
alter index <schema>.METRIC_DATA_6D_1S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_7D_0S initrans 15;
alter index <schema>.METRIC_DATA_7D_0S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_7D_1S initrans 15;
alter index <schema>.METRIC_DATA_7D_1S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_8D_0S initrans 15;
alter index <schema>.METRIC_DATA_8D_0S_MID_IDX initrans 15;
alter table <schema>.HQ_METRIC_DATA_8D_1S initrans 15;
alter index <schema>.METRIC_DATA_8D_1S_MID_IDX initrans 15;
```

Restart Hyperic Server

Restart the Hyperic Server.

Periodic Oracle Database Maintenance

Perform the following maintenance after running Hyperic for about a week, and monthly thereafter:

1. Run the `table_maintenance.sql` procedure attached to this page to compact and shrink space on all Hyperic tables (except `HQ_METRIC*` tables).
2. Run the `rebuild_index.sql` procedure attached to this page to rebuild the index on all Hyperic tables.

3. Run tablespace maintenance as appropriate:

- a. Perform the following query to determine how fragmented a tablespace is. The query returns the tablespace name and total count of holes in it.

```
select count(*), tablespace_name
from dba_free_space
group by tablespace_name
order by 1,2;
```

- b. If the hole count is in the order of thousands, run **Reorganize**.

The screenshot displays the Oracle Enterprise Manager 11g Database Control interface for tablespace **TS_HQDB**. The page shows various configuration details such as Name, Bigfile tablespace status, Status (ReadWrite), Type (Permanent), Extent Management (local), Encryption (NO), Storage (Automatic), and Segments Space Management (Automatic). A table of Datafiles is shown with columns for Name, Directory, Size (MB), and Used (MB). Below the table, the 'Tablespace Full Metric Thresholds' section shows Space Used (%) at 97, Warning (%) at 85, and Critical (%) at 97. The 'Free Space (MB)' section shows Warning (MB) and Critical (MB) as 'Not Defined'. An arrow points to the 'Reorganize' button in the Actions menu at the top right of the configuration area.

Name	Directory	Size (MB)	Used (MB)
ts_hqdb_reorg0	/array/oracle/oradata/orcl/	32,767.98	571.11
ts_hqdb2_reorg0	/array/oracle/oradata/orcl/	25,600.00	518.38

Set Up PostgreSQL

- [Set Up External Hyperic Database on PostgreSQL \(see page 49\)](#)
 - [Example Configuration \(see page 49\)](#)
 - [Install and Initialize PostgreSQL \(see page 49\)](#)
 - [Create PostgreSQL User \(see page 50\)](#)
 - [Configure PostgreSQL Server Options \(see page 50\)](#)
 - [PostgreSQL Tuning for Large Deployments \(Optional\) \(see page 50\)](#)
 - [Configure Client Authentication \(see page 51\)](#)
 - [Install PostgreSQL Client on the Hyperic Server Host \(see page 51\)](#)
 - [Install the Hyperic Server \(see page 51\)](#)
 - [Start Hyperic Server \(see page 51\)](#)
- [Troubleshoot PostgreSQL Connection Problems \(see page 52\)](#)
- [Useful PostgreSQL Commands \(see page 52\)](#)

Set Up External Hyperic Database on PostgreSQL

This section provides instructions for setting up PostgreSQL as your external Hyperic database.

This task corresponds to [Step 2 – Set Up Hyperic Database \(see page 20\)](#) of [Hyperic Installation and Startup Process \(see page 20\)](#).

It is assumed that the reader is or has the support of a database administrator.

Evaluating Hyperic?

If you are installing Hyperic for evaluation, you can use Hyperic's built-in PostgreSQL database, rather than set up an external database.

These instructions assume that you are performing a new installation of an RPM package of PostgreSQL, using Yum, an automatic RPM package installer.

If you do not have an RPM package, you can obtain the source from the "<http://www.postgresql.org/>"

Hyperic requires utf-8 character encoding

Create the Hyperic database with utf-8 character encoding.

Example Configuration

The instructions that follow show commands for setting up PostgreSQL in an environment with these characteristics:

Operating system	Red Hat Enterprise Linux 4
Database	PostgreSQL 8.3
Database IP address	192.168.1.4
Hyperic Server IP address	192.168.1.6
Database user name	admin
Database password	hqadmin
Database location	/var/lib/pgsql/data/
PostgreSQL configuration file	/var/lib/pgsql/data/postgresql.conf
PostgreSQL authorization file	/var/lib/pgsql/data/pg_hba.conf

Install and Initialize PostgreSQL

Log in as root to the target Linux machine, and enter this command:

```
yum install postgresql postgresql-server
```

PostgreSQL will be installed in:

```
/etc/init.d/  
/usr/bin/  
/usr/share/doc/  
/var/lib/pgsql/
```

Create PostgreSQL User

1. Change user to PostgreSQL and connect to the database locally.

```
# su postgres
```

- The psql prompt is displayed.

2. Create a user named admin with login and createdb privileges.

```
create role admin with login createdb password 'hqadmin';
```

3. Create a default database for Hyperic. Place quotes around the string HQ so that the database name will be uppercase.

```
CREATE DATABASE "HQ" OWNER admin;
```

Configure PostgreSQL Server Options

In this step, you configure PostgreSQL Server options in the PostgreSQL server configuration file, `postgresql.conf`.

1. Open `postgresql.conf`.
2. The default database permissions allow local connections only. To configure PostgreSQL to listen on all network interfaces, uncomment the listen address entry and change its value as shown below.

```
listen_addresses = '*'
```

3. Add the following settings to optimize Hyperic performance:

```
##performance changes for HQ  
shared_buffers=10000  
work_mem=2048  
statement_timeout=30000
```



In PostgreSQL 8.3, the PostgreSQL parameter that enables Hyperic to monitor the database — `track_counts` — is enabled, because the PostgreSQL autovacuum daemon needs the collected information. Only superusers can change this setting.

PostgreSQL Tuning for Large Deployments (Optional)

These changes and additions to `postgresql.conf` can improve Hyperic performance in large environments, if you have at least 2GB RAM available for the database.

```
shared_buffers = 20000
commit_delay = 10000
checkpoint_segments = 15
work_mem = 8192
maintenance_work_mem = 32768
max_fsm_pages = 40000
effective_cache_size = 5000
```

In particular, increasing `effective_cache_size` is beneficial, given sufficient RAM.

You may need to refine your database configuration based on the performance you experience. Review of the database log by a database administrator should indicate whether further adjustments to `checkpoint_segments` or `max_fsm_pages` are appropriate.

Configure Client Authentication

In this step, you configure PostgreSQL to allow connections from other users and from the Hyperic Server.

PostgreSQL client authentication is defined in the `pg_hba.conf` file, which contains lines, referred to as records, that specify allowed connection types, users, client IP addresses, and authentication method. Locate this line in the file:

```
# TYPE DATABASE USER CIDR-ADDRESS AUTH-METHOD
```

and add these lines below it:

```
local all all ident sameuser
host all all *192.168.1.6/32* password
```

For more information about `pg_hba.conf` see "<http://www.postgresql.org/docs/8.2/interactive/auth-pg-hba-conf.html>"

Install PostgreSQL Client on the Hyperic Server Host

Install the PostgreSQL client so that you can verify connectivity between the server and the database. To install the client, enter these commands:

```
yum -y install postgresql
psql -d postgres -h 192.168.1.4 -U admin -W
```

After you verify the connection, you can remove the PostgreSQL client with this command:

```
yum -y remove postgresql
```

Install the Hyperic Server

Install the Hyperic Server. For instructions, see [Step 3 - Set Up Hyperic Server \(see page 22\)](#) of the [Hyperic Installation and Startup Process \(see page 20\)](#).



Don't start up the server yet

Do not start the Hyperic Server until after completing the steps in the following section.

Start Hyperic Server

Start the Hyperic Server. For instructions, see the ["Start the Hyperic Server" step \(see page 23\)](#) of the [Hyperic Installation and Startup Process \(see page 20\)](#).

If the server fails to start up, there may be problems with your PostgreSQL configuration. Check the PostgreSQL logs for connection failures or errors.

Troubleshoot PostgreSQL Connection Problems

If network connections to the database fail, you can troubleshoot the issue in PostgreSQL log files, using the UNIX® `tail` command with the `-f` parameter

`tail -f` displays the lines at the end of a file, and displays additional log messages that follow to the terminal. This is useful for watching log files, or any other file which may be appended over time. Failed connection messages are written to the following files:

- `/var/lib/pgsql/data/pg_log/postgresql-day.log`
- `/var/lib/pgsql/pgstartup.log`

Useful PostgreSQL Commands

- `\h` — help with SQL commands
- `?` — help with psql commands
- `\du` — list roles/users
- `\l` — list databases
- `\c` — to choose a database
- `\d` — to list tables once in a database
- `\q` — quit

Run Hyperic Installer

- [Obtain Hyperic Installer \(see page 53\)](#)
- [About the Setup Script \(see page 53\)](#)
- [Run the Setup Script \(see page 54\)](#)

About this page...

*This page has instructions for performing a **new** installation of Hyperic components using the Hyperic installer. Before running the installer, see [Hyperic Installation and Startup Process \(see page 20\)](#) for information about where installation fits in the implementation process. For production environments, there are several steps to perform before installing the Hyperic Server and Hyperic Agent, including database setup and SSL configuration.*

If you wish to upgrade an existing Hyperic deployment, please see [Hyperic Upgrade Processes \(see page 28\)](#).

Obtain Hyperic Installer

If you have not done so already, download the Hyperic installer. See [Select and Download an Installer \(see page 31\)](#).

About the Setup Script

The setup script, `setup.bat` for Windows or, `setup.sh` for Unix-like environments, is in the Hyperic installation package. You can use to install the Hyperic Server, the Hyperic Agent, or both.

When you run the setup script, you can supply a qualifier that sets the installation mode.

- `none` — Run the setup script with no qualifier to perform a quick install. Selected components will be installed with default values for most configuration options. If you install the Hyperic Server, it will be configured to use its built-in PostgreSQL database. A quick install is useful when you evaluate Hyperic.
- `-full` — In "full" mode, the installer dialog will prompt for all setup configuration options. Use this option if you plan to configure the Hyperic Server to use an SSL keystore that you manage yourself, rather than a Hyperic-generated keystore.
- `-upgrade` — Use "upgrade" mode if you have an existing Hyperic Server installation. The server configuration and the contents of the existing Hyperic database are preserved. See [Hyperic Upgrade Processes \(see page 28\)](#) before doing an upgrade installation.
- `-postgresql` — Quick install when using a standalone (not the Hyperic built-in) PostgreSQL database; installer will prompt you for database connection information and use defaults for other configuration settings.
- `-oracle` — Quick install mode for Oracle; installer will prompt you for database connection information and use defaults for other configuration settings.
- `-mysql` — Quick install mode for MySQL; installer will prompt you for database connection information and use defaults for everything else.
- `-updateScale` — In vFabric Hyperic 4.6.5 and later, run the installer with this option to change the sizing profile for the Hyperic Server. For information about sizing profiles, see [About Sizing Profiles in vFabric Hyperic \(see page 18\)](#). For information about changing the sizing profile for vFabric Hyperic Server, see [Change vFabric Hyperic Server Sizing Profile \(see page 111\)](#).

Configuration Options for Database-Specific Quick Installs

When you do a database-specific quick install (`-postgresql`, `-oracle`, or `-mysql`) the installer **does not**:

- Prompt for the ports on which the Hyperic Server listens for agent and user interface requests — the default ports are automatically configured.
 - Allow you to specify the location and password for a user-managed server keystore — instead, the Hyperic Server will use a Hyperic-managed keystore.
 - Allow you to select a different database type.
-

Run the Setup Script

This section describes the dialog that the installation script presents if you run it in `-full` mode. If you use a different mode, some of the prompts described below will not appear — default values will be applied to those options.

1. Create a directory for the Hyperic installation.
 - The installation dialog assumes your Hyperic installation directory is `/home/hyperic` on Unix-like systems and `c:\Program Files` on Windows.
2. Unpack the tarball or zip archive.
 - **Use GNU Tar to unpacking Hyperic tarballs.** Use of proprietary Unix Tar utilities will result in warnings. GNU Tar is available at "<http://www.gnu.org>"

```
tar zxvf hyperic-hq-installer-4.x.y-xxx.tgz
```

- **If you are installing Hyperic components on a Windows platform, you must run the installer on a local drive.**
3. Open a command shell.
 - On Unix-based platforms, enter:

```
PathToInstaller/setup.sh -mode
```


- On Windows platforms, enter:


```
PathToInstaller\setup.bat -mode
```

- where *mode* is one of the values in the table above, with the exception of `upgrade` — if you are upgrading an existing Hyperic deployment, see [Hyperic Upgrade Processes \(see page 28\)](#).
4. The VMware License agreement is displayed.
 - Accept the license agreement to proceed.
 5. **Choose which software to install 1: Hyperic HQ Server 2: Hyperic HQ Agent**
 - To install both the server and the agent, enter:

```
1,2
```

6. **HQ server installation path [default '.....']:**
 - Accept the default, or enter a directory location. You must have write access the location.
7. **What port should the HQ server's web-based GUI listen on for http communication? [default '7080']:**
 - Accept the default, or enter a different (unused) port.
8. **What port should the HQ server's web-based GUI listen on for secure https communication? [default '7443']**
 - Accept the default, or enter a different (unused) port.
9. **Would you like us to use your own java keystore? [default '2'] 1: Yes 2: No**
 - Enter "1" if you wish to configure the Hyperic Server to use a certificate you manage, rather than generate its own, and proceed to the next step.
 - If you accept the default "2", a default keystore will be generated at:
ServerHome/conf/hyperic.keystore with the password hyperic, and the next three prompts will not appear.
10. **What is the file path to your java keystore?**
 - This prompt appears if you are configuring Hyperic Server for a user-managed keystore.
 - Enter the path to your keystore.
11. **What is the file password to your java keystore?**
 - This prompt appears if you are configuring Hyperic Server for a user-managed keystore.
 - Enter the path to your keystore.
12. **Enter the base URL for the Hyperic server's web-based GUI [default...]**
 - The URL used to access the Hyperic Server. This value is used in alert notification emails. This value can be changed on the **Administration** page in the Hyperic Portal.
13. **Enter the fully qualified domain name of the SMTP server that Hyperic will use to send email messages [default FQDN.local]**
 - If the installer does not find a local SMTP server, and you do not specify one, Hyperic cannot send alert notifications. Other alert functionality is unaffected.
 - You can configure Hyperic for an external SMTP server after installing Hyperic Server. See [Configuring Hyperic Server for SMTP Server](#).
14. **Enter the email address that HQ will use as the sender for email messages [default...]**
 - The email address of the Hyperic Administrator. Note that most mail servers will not deliver mail without a valid domain name in the From field.
15. **What is the installation profile? default '1':]**
 - 1: small (less than 50 platforms)**
 - 2: medium (50-250 platforms)**
 - 3: large (larger than 250 platforms)**
 - Select the profile that fits your deployment. For more information, see [About Sizing Profiles in vFabric Hyperic \(see page 18\)](#).

 The sizing prompt is only presented in vFabric Hyperic, and only when you are running a `-full` installation.

 If you plan to use the embedded PostgreSQL database, rather than an external database (required for production environments), select the "small" sizing profile. The "medium" and "large" profiles are *not* supported unless you use an external Hyperic database.

16. **What backend database should the HQ server use? [default '1']:Choices:**
- 1: **HQ Built-in Database**
 - 2: **Oracle 10g/11g**
 - 3: **PostgreSQL**
 - 4: **MySQL Enterprise / Community Server 5.x**
 - Select the database you wish to use and:
 - If you entered "1", proceed to step 16.
 - If you entered "2", "3", or "4", proceed to step 17.
17. **What port should HQ's built-in database use? [default '9432']:**
18. Dialog for external database:
- a. **Enter the JDBC connection URL.**
 - The prompt supplies a default URL, which assumes the external database is on localhost.
 - Supply a URL in the form shown, editing as appropriate to identify the connection details, such as host and name.
 - b. **Enter the username to use to connect to the database:**
 - Enter the database username that was set up when the Hyperic database was created.
 - c. **Enter the password to use to connect to the database:**
 - Enter the database password that was set up when the Hyperic database was created.
19. **Would you like to use an auto generated encryption key to encrypt the database password? [default '1']**
- If you accept the default, the installer will generate a key for encrypting the database password. Enter 2 if you prefer to supply the string yourself, and when prompted, enter a string of at least 8 characters.
20. **What should the username be for the initial admin user? [default 'hqadmin']:**
- Accept the default, or enter a different username.
21. **What should the password be for the initial admin user?:**
- The installer will not echo the password but will prompt for it twice.
22. **What should the email address be for the initial admin user? [default...]**
- Accept the default, or enter a different email address.
23. If the installer detects a database from a previous Hyperic installation, it will prompt you to:
- **1: Upgrade the HQ server database** — Choose this option to preserve your existing Hyperic data, then follow the instructions in [Upgrade Hyperic Server \(see page 106\)](#).
 - **2: Overwrite the HQ server database** — This option erases all of the data in your Hyperic database. If you do not want to preserve the contents of the database, it would be quicker to drop and recreate the database prior to installing Hyperic Server.
 - **3: Exit the installer**
24. **HQ agent installation path [default '.....']:**
- Accept the default, or enter a different path.
 - The installer indicates the installation was successful, provides the URL for the Hyperic Portal along with the default username and password, and returns you to the command prompt.

After installing the Hyperic Server, proceed with license installation (for vFabric Server only) and start the server, as described in the [Install License \(see page 22\)](#) and [Start the Hyperic Server \(see page 23\)](#) steps of [Hyperic Installation and Startup Process \(see page 20\)](#).

Install an Agent-Only Package

- [Install from Agent Tarball \(see page 57\)](#)
- [Install from Agent Zip Archive \(see page 57\)](#)

About this page...

*This page has instructions for performing a fresh agent installation from an agent-only archive — they correspond to the **Install the Agent** task in [Step 4 - Set Up the Hyperic Agent \(see page 24\)](#) step of the [Hyperic Installation and Startup Process \(see page 20\)](#). Before installing the agent, review the agent setup instructions make sure that you have performed prerequisite steps appropriate for your environment, such as SSL setup and JRE configuration.*

If you have multiple agents to install, see [Install Hyperic Agents in Volume \(see page 100\)](#).

Install from Agent Tarball

On non-Windows systems, the Hyperic Agent is automatically installed as a daemon.

1. Create a directory for the Hyperic Agent.
2. Unpack the tarball into the agent directory.
 - Starting the agent will run it as a daemon process.



Unpack Tarballs with GNU Tar Only

Use GNU Tar to unpacking Hyperic tarballs. Use of proprietary Unix Tar utilities will result in warnings. GNU Tar is available at "<http://www.gnu.org>"

After installing the agent, proceed to the next step in [Step 4 - Set Up the Hyperic Agent \(see page 24\)](#).

Install from Agent Zip Archive

To install the Hyperic Agent as a Windows Service on a Windows system:

1. Create a directory for the Hyperic Agent.
2. Unpack the archive into the agent directory.

After installing the agent, proceed to the next step in [Step 4 - Set Up the Hyperic Agent \(see page 24\)](#).

Run Windows MSI Installer

- [About the Hyperic MSI Installer \(see page 57\)](#)
- [Interactive MSI Installation \(see page 58\)](#)
- [Silent MSI Installation \(see page 66\)](#)
- [Remote Installs with Microsoft Management Console \(see page 71\)](#)
- [Solving Service Startup Problems After MSI Install \(see page 71\)](#)

About this page...

This page has instructions for installing Hyperic components using the Windows MSI installer. As noted below in [About the Hyperic MSI Installer \(see page 57\)](#), the MSI installer is most suitable for non-production environments. For information about other installers, see [Select and Download an Installer \(see page 31\)](#).

instructions on installing Hyperic for production use, see [Hyperic Installation and Startup Process \(see page 20\)](#).

About the Hyperic MSI Installer


You can use the Hyperic MSI Installer to install the Hyperic Server, the Hyperic Agent, or both. Both server and agent are installed as Windows services.

Note that the MSI installer installs Hyperic Server with a built-in PostgreSQL database. You cannot install and configure the Hyperic Server to use an external database with the MSI installer. For this reason, the MSI installer is most useful for quickly setting up a Hyperic evaluation environment.

The Hyperic MSI installer provides an InstallShield wizard GUI that has two modes:

- Complete — When you select the "Complete" installation option, the installation location is the *only* user input the wizard allows — it will install both the server and the agent and use default values for all other installation options. See [MSI Silent Mode Properties \(see page 67\)](#) to see the default values for server and agent configuration settings.
- Custom — The wizard's "Custom" mode allows you more control over the installation. You can install either the server or the agent, or both. In addition, you can explicitly specify these options:
 - Address of SMTP server the Hyperic Server will use to send email
 - Username and password for the initial admin user.
 - The encryption key used to encrypt the database user password.
 - The agent address and listen port.

You can also run the the MSI installer in silent mode, in which case you supply installation properties — defined in [MSI Silent Mode Properties \(see page 67\)](#) — on the command line. The default values are used for properties that are not defined at the command line.

 Requirements for MSI Installer

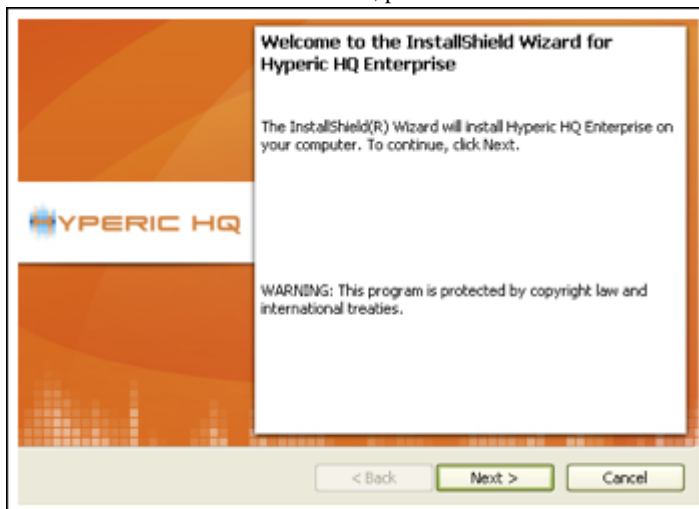
Note that:

- To run the MSI Installer you must have administrator privilege and permission to install Windows Services.
- Only the user that installed Hyperic with the MSI installer can uninstall Hyperic with the MSI installer.

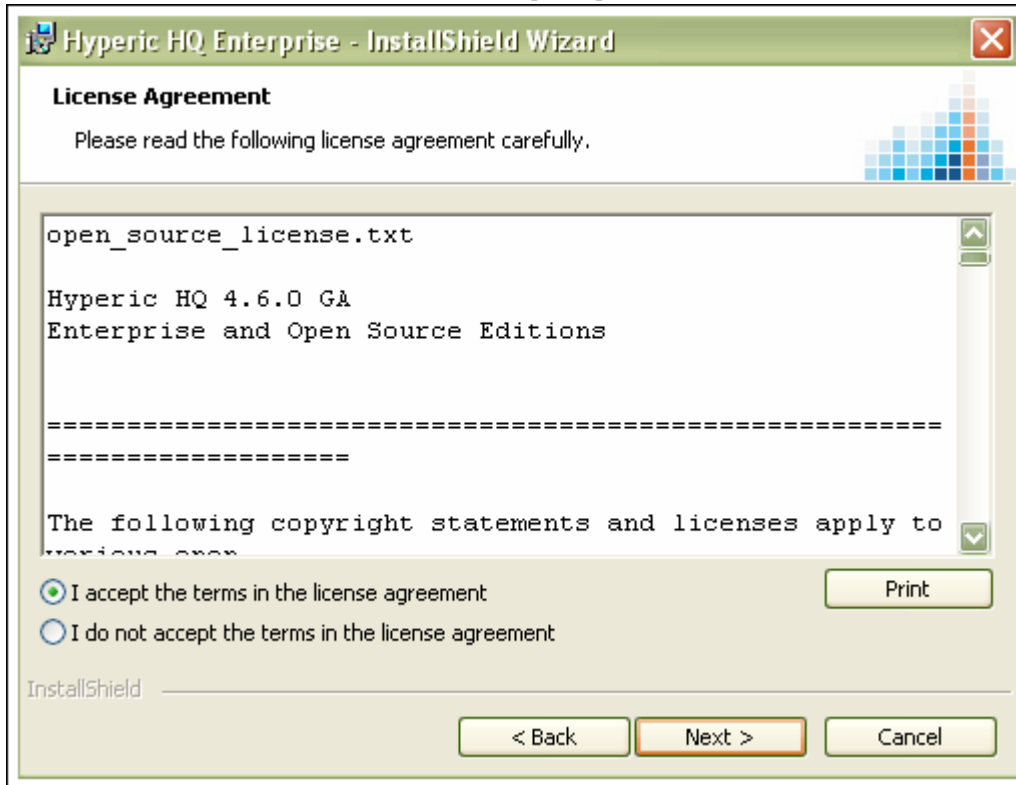
Interactive MSI Installation

Interactive Complete Install

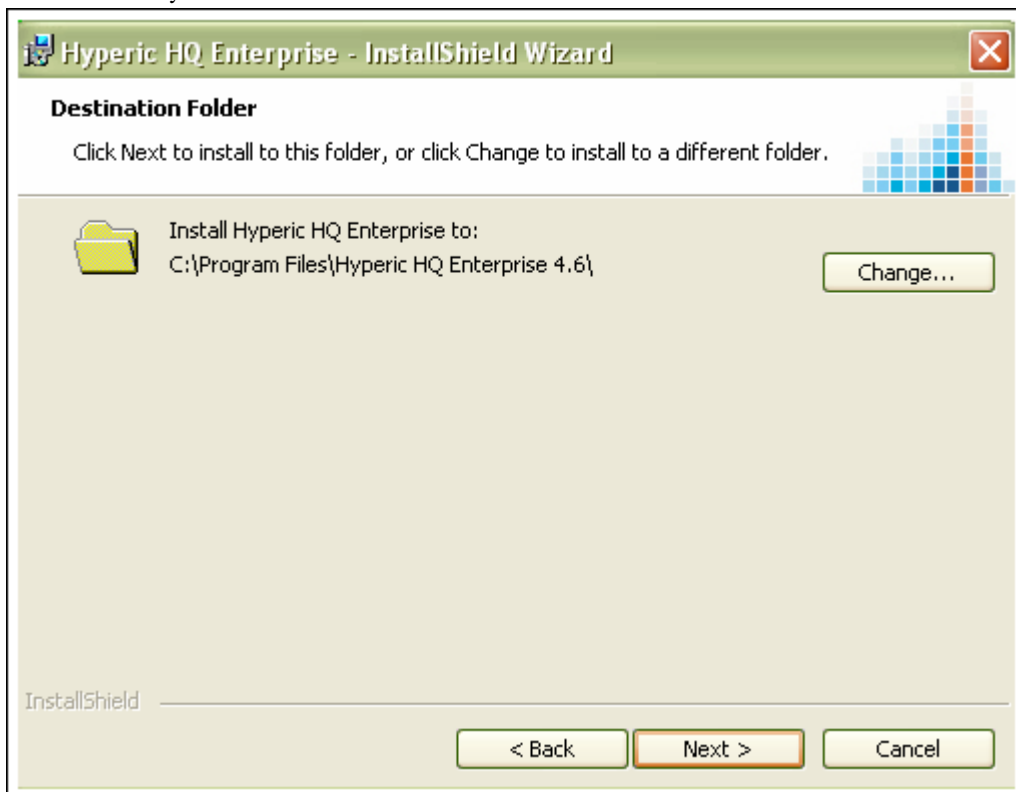
1. Double-click the Hyperic MSI installer file.
2. On the **InstallShield Wizard** window, press **Next** to start the installation.



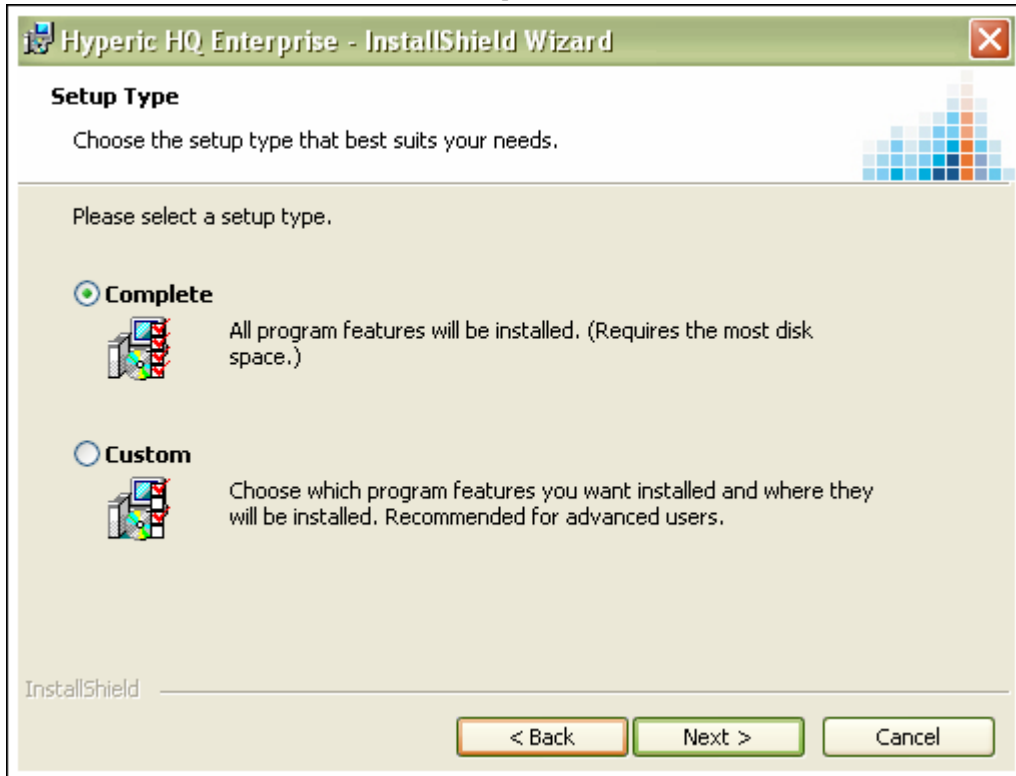
3. On the **License Agreement** window, select the "I accept..." option and click **Next**.



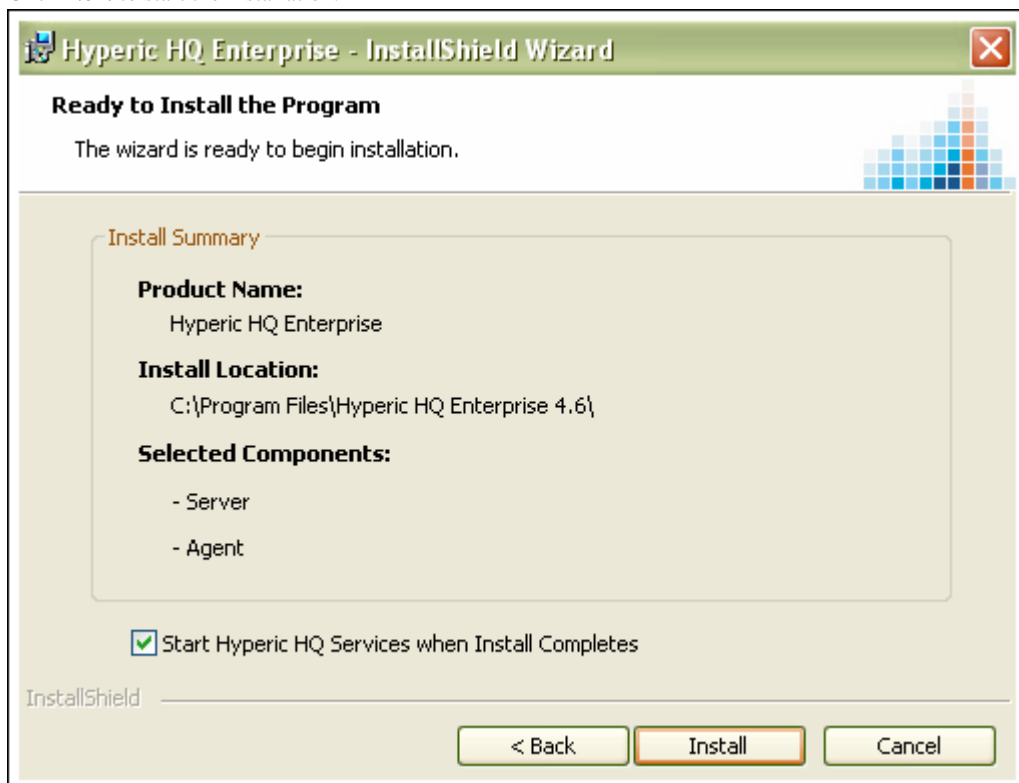
4. On the **Destination Folder** window, click **Next** to install to the default installation directory, or click **Change** to select a different directory.



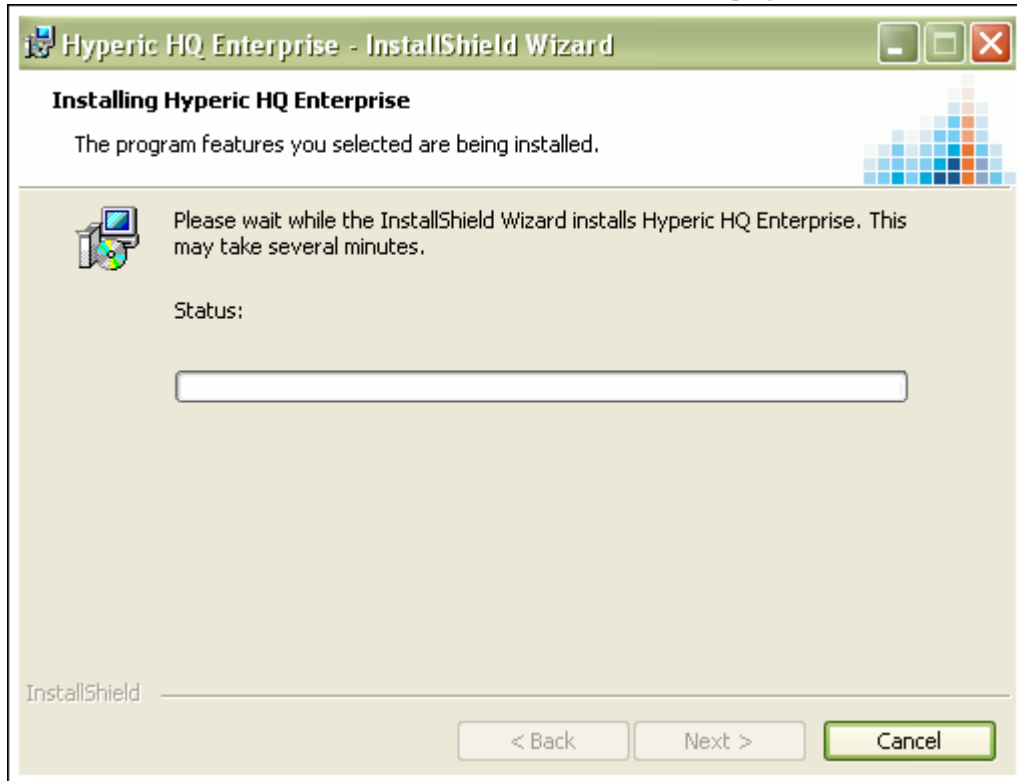
5. On the **Setup Type** window, select the **Complete** option and click **Next**.



6. On the **Ready to Install** window:
- Uncheck the box if you do not want the services for the Hyperic Server and Hyperic Agent to be started when the installation is complete.
 - Click **Next** to start the installation.



7. On the **Installing Hyperic Enterprise** window, the "Status" bar shows the progress of the installation.



8. Click **Next** when the installation is complete.

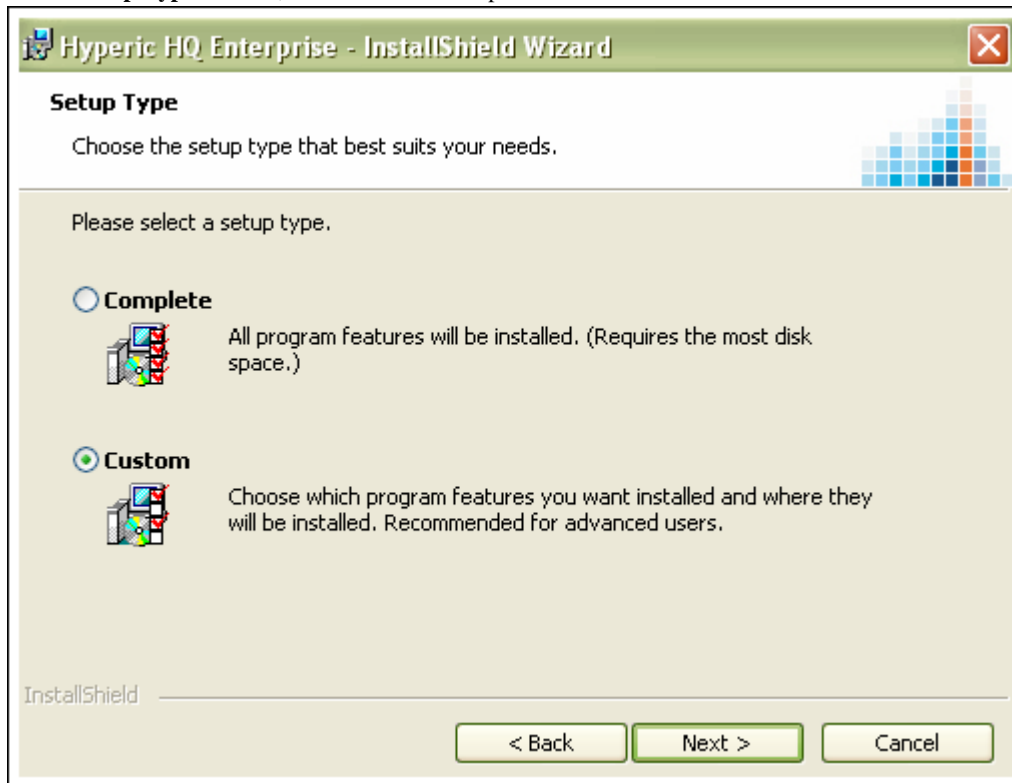
Interactive Custom Install

The custom installation procedure allows you to both the Hyperic Server and the Hyperic Agent, or just one or the other.

To run the custom install procedure:

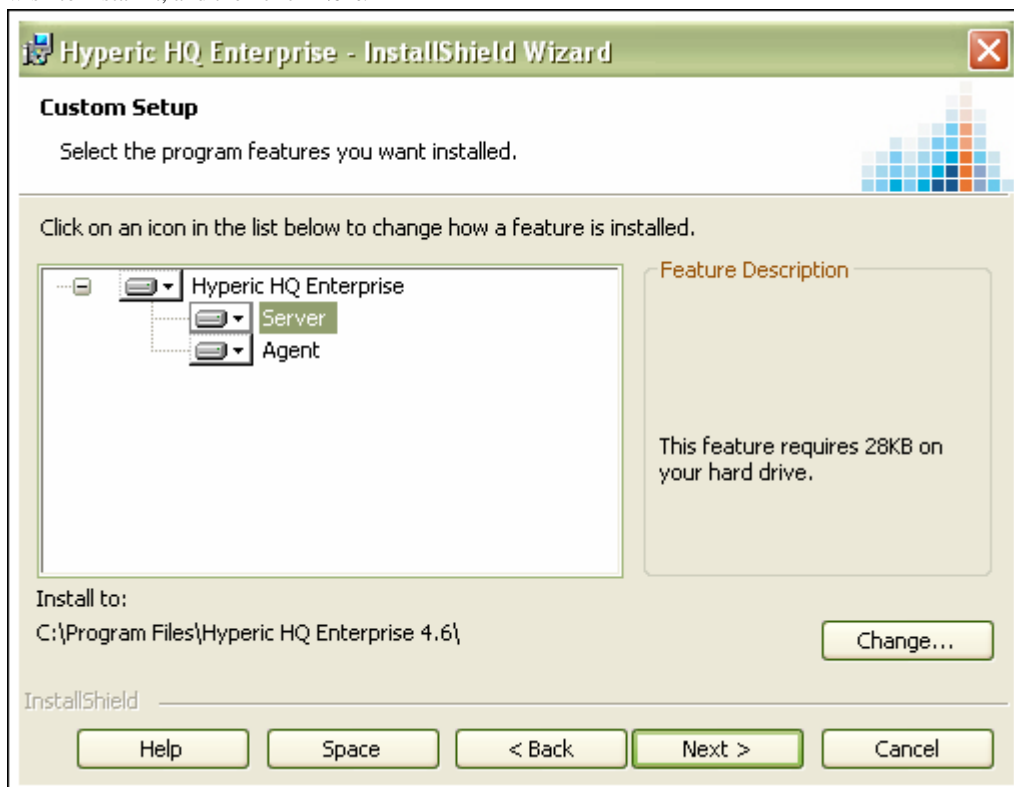
1. Perform steps 1 through 4 of the [Interactive Complete Install \(see page 58\)](#) above.

2. On the **Setup Type** window, Select the **Custom** option and click **Next**.



3. On the **Custom Setup** window you can:

- Click **Next** to install both server and agent to the installation directory shown, or
- Click **Change** to select a different installation directory and then click **Next**, or,
- Expand the **Hyperic Enterprise** tree control so you can deselect the **Server** or **Agent** component if you do not wish to install it, and then click **Next**.



4. On the **HQ Web Configuration** window, you can accept the default value, or enter a different value for:
- **Address** — The address upon which the Hyperic Server will listen for agent and web application requests. The default value is the IP address of the first network interface the installer found on the platform.
 - **HTTP Port** — The port upon which the Hyperic Server will listen for plain text agent and web application requests. The default is port 7080.
 - **HTTPS Port** — The port upon which the Hyperic Server will listen for SSL agent and web application requests. The default is port 7443.

Hyperic HQ Enterprise - InstallShield Wizard

HQ Web Configuration
Enter your Hyperic HQ Web Configuration here

Address

HTTP Port

HTTPS Port

Note: the Hyperic HQ Dashboard will be accessible via the URL of the form `http://{HOST}:{PORT}/`

InstallShield

< Back Next > Cancel

5. On the **Mail Server Configuration** window, accept the defaults or change the values for:
 - SMTP Host — The hostname of the SMTP server found on the platform.
 - Mail Sender — The email address from which the server will send email notifications.

Hyperic HQ Enterprise - InstallShield Wizard

Mail Server Configuration

Enter your Mail Server Configuration here

SMTP Host: Roger

Mail Sender: hqadmin@Roger

InstallShield

< Back Next > Cancel

6. On the **Database Encryption Key** window, enter a string of at least eight characters.

Hyperic HQ Enterprise - InstallShield Wizard

Database Encryption Key

Enter an encryption key for encrypting the database password

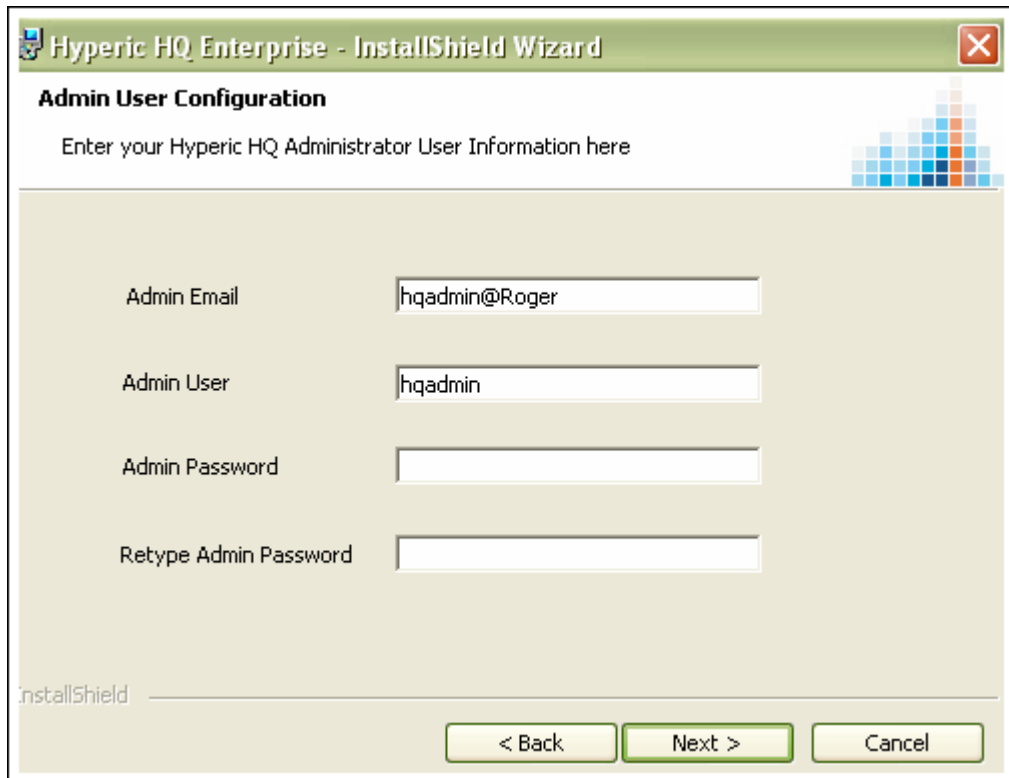
Encryption Key

InstallShield

< Back Next > Cancel

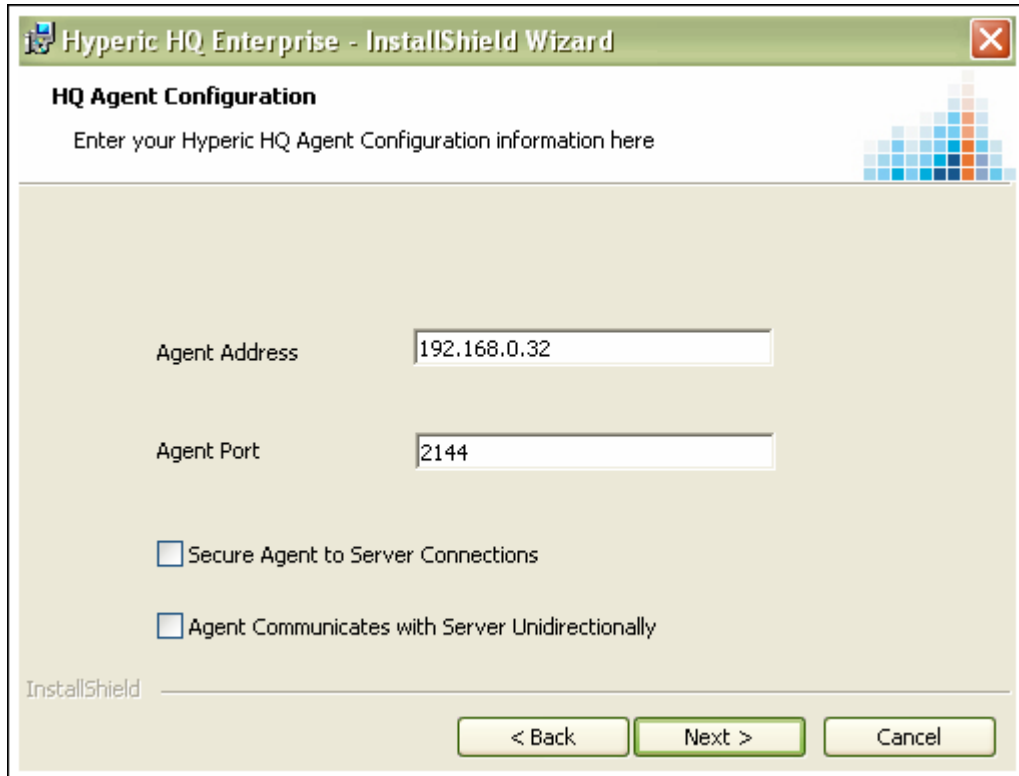
7. On the **Admin User Configuration** window, accept the defaults or change the values, for:

- Admin Email
- Admin User
- Admin Password



The screenshot shows a window titled "Hyperic HQ Enterprise - InstallShield Wizard" with a close button in the top right corner. The main heading is "Admin User Configuration" and the instruction is "Enter your Hyperic HQ Administrator User Information here". There are four input fields: "Admin Email" with the value "hqadmin@Roger", "Admin User" with the value "hqadmin", "Admin Password" (empty), and "Retype Admin Password" (empty). At the bottom, there are three buttons: "< Back", "Next >" (highlighted in green), and "Cancel". The "InstallShield" logo is visible in the bottom left corner.

8. On the **HQ Agent Configuration** window, accept the defaults or change the values for:
- Agent Address — This is the address the Hyperic Server will use to contact the agent.
 - Agent Port — This is the port the Hyperic Server will use to contact the agent.
 - Secure Agent to Server Connections — If you wish all communication between agent and server to be SSL.
 - Agent Communicates with Server Unidirectionally — If you wish the agent to initiate all communication with the server.

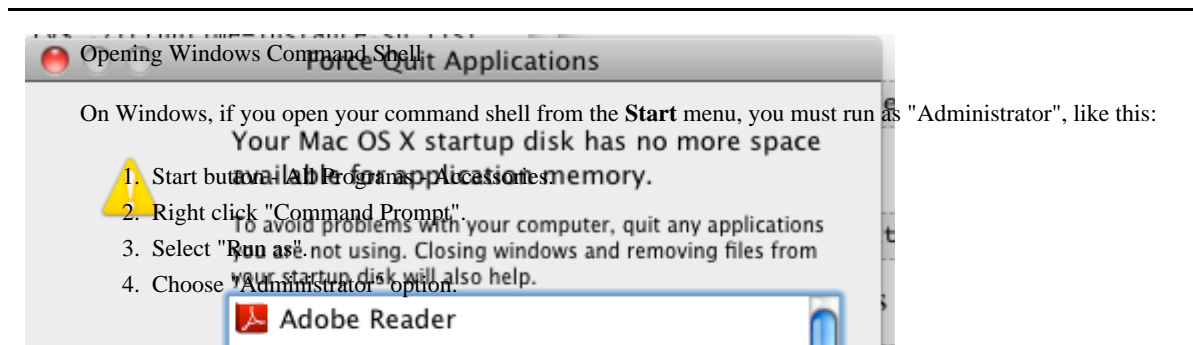


9. On the **Ready to Install the Program** window:
- Deselect "Start Hyperic HQ Services when Install Completes" option if you **do not** want services for the Hyperic Server and Hyperic Agent to be started when the installation is complete.
 - Click **Install** to install selected components.

Silent MSI Installation

You can run the Hyperic MSI installer in silent mode from the DOS prompt.

Command Line Syntax



To install Hyperic:

```
%Comspec% /c msexec /i "PathToInstaller" /qn PROPERTY1=VALUE1 PROPERTY2=VALUE2 ...
```

Where:

- /c — tells the shell to wait until the `msiexec.exe` command completes before proceeding. Without this switch, the shell will execute subsequent commands before the current command finishes.
- /i — instructs MSIEEXEC to install the .MSI listed after the switch.
- PathToInstaller — is the full path to the MSI installer, for example `C:\Program Files\hyperic-hq-installer-4.6.msi`
- /qn — turns off prompts to the console.
- PROPERTY1=VALUE1 — is a installation property name=value pair, for example `INSTALLDIR=C:\hyperic`. Installation properties are defined below in [MSI Silent Mode Properties \(see page 67\)](#)

To uninstall:

```
%Comspec% /c msiexec /x "PathToInstaller" /qn
```

- /x — instructs MSIEEXEC to uninstall the .MSI listed after the switch.

MSI Silent Mode Properties

Installation properties for installing Hyperic in MSI silent mode are described below.

Note that all properties and their values are case-sensitive.

Properties that begin with the strings `SERVER` and `AGENT` are server and agent properties, respectively.

Silent Mode Property	Description	Default
INSTALLDIR	Directory where the Hyperic components will be installed.	C:\Program Files'
ADDLOCAL	Use this property if you want to install only the Hyperic Server or the Hyperic Agent. Comma-separated list of components to be installed. Allowable case-sensitive values are Agent and Server.	If you do not spec server will be ins
SERVER_ADMIN_EMAIL	Hyperic Server Administrator's email address.	SERVER_ADMI
SERVER_ADMIN_USER	User name of the original admin user in Hyperic.	hqadmin
SERVER_ADMIN_PASSWORD	Password of the original admin user in Hyperic Server.	hqadmin
SERVER_DATABASE_USER	Defines the username the Hyperic Server will use when connecting to the Hyperic database. The value assigned will be saved in the <code>server.database-user</code> property in <code>server.conf</code> .	hqadmin
SERVER_DATABASE_PASSWORD	Password the Hyperic Server will use when connecting to the Hyperic database. The value assigned will be saved in the <code>server.database-password</code> property in the <code>server.conf</code> file.	hqadmin

DB_ENC_KEY_PW	The Database Encryption Key Password. This value will be used to encrypt the database password. Must be at least 8 characters long, there are no restrictions on the type of characters that may be entered.	
SERVER_MAIL_HOST	The IP address or hostname of the SMTP server that the Hyperic Server will use for sending alerts and other Hyperic-related emails. Most UNIX platforms have a local SMTP server. If you wish to use a non-local SMTP server, specify the address with this property. The value assigned will be saved in the <code>server.mail.host</code> property in <code>server.conf</code> file.	127.0.0.1
SERVER_MAIL_SENDER	The 'From' address in email notifications from the Hyperic Server.	SERVER_ADMIN
SERVER_POSTGRES_PORT	Hyperic Server's embedded database listen port.	9432
SERVER_WEBAPP_HOST	Specifies the Hyperic Server's listen address for Hyperic Portal communications. By default, this property and <code>AGENT_SERVER_ADDRESS</code> have the same value. If you wish, you can use these properties to designate different hosts for agent-server and agent-portal communications. The value assigned will be saved in <code>server.webapp.host</code> in <code>server.conf</code> .	host IP address
SERVER_WEBAPP_PORT	Specifies the Hyperic Server listen port on which the server listens for Hyperic Portal communications in non-secure mode. By default, this property and <code>AGENT_SERVER_PORT</code> have the same value. If you wish, you can use these properties to designate different ports for agent-server and agent-portal communications. The value assigned will be saved in the <code>server.webapp.port</code> property in <code>server.conf</code> .	7080
SERVER_WEBAPP_SECURE_PORT	Specifies the Hyperic Server port on which the server listens for Hyperic Portal communications in secure mode. The value assigned will be saved in the <code>server.webapp.secure.port</code> property in <code>server.conf</code> .	7443
HQ_START_SERVICES	Property indicating whether the Hyperic Agent and Server processes should be started as a Windows Service at the end of the installation. 1 indicates true, 0 indicates false.	1

AGENT_ADDRESS	The IP address to which the agent binds at startup. The default value allows the agent to listen on all IP addresses on the the agent host. The value assigned is saved in both the <code>agent.listenIp</code> and the <code>agent.setup.agentIP</code> properties in <code>agent.properties</code> . If there is a firewall between the agent and the server, set AGENT_ADDRESS to the firewall address. After installation is complete, set <code>agent.listenIP</code> in <code>agent.properties</code> to the agent's local IP address, and configure the wall to forward agent-bound traffic to that address.	host IP address
AGENT_IS_SECURE	Indicates whether communications between the Hyperic Agent and the Hyperic Server should take place over a secure encrypted channel. The setting will be stored appropriately in the <code>agent.setup.camSecure</code> property in <code>agent.properties</code> . <ul style="list-style-type: none"> ▪ 1 indicates secure communications ▪ 0 indicates that communications will not be secured. 	0
AGENT_PORT	The port on the agent's listen address to which the agent binds at startup. This value is saved to both <code>agent.setup.agentPort</code> and <code>agent.listenPort</code> in <code>agent.properties</code> .	2144
AGENT_SERVER_ADDRESS	Specifies the IP address the agent connects to to reach the Hyperic Server. The value is saved to <code>agent.setup.camIP</code> in <code>agent.properties</code> .	host IP address
AGENT_SERVER_USER	The Hyperic username the agent will use when it registers with the server. The value is saved to <code>agent.setup.camLogin</code> in <code>agent.properties</code> . Typically this property and AGENT_SERVER_PASSWORD have the same values as SERVER_ADMIN_USER and SERVER_ADMIN_PASSWORD respectively. However, if you are installing a server and an agent on the same host, and the agent will report to a server on a different host, you might specify different credentials for AGENT_SERVER_USER/AGENT_SERVER_PASSWORD and SERVER_ADMIN_USER/SERVER_ADMIN_PASSWORD.	hqadmin
AGENT_SERVER_PASSWORD	The password for the user specified by AGENT_SERVER_USER. The value is saved to <code>agent.setup.camPword</code> in <code>agent.properties</code> .	hqadmin
AGENT_SERVER_PORT	Port on server port to use for non-secure communications with the server. The value is saved to <code>agent.setup.camPort</code> in <code>agent.properties</code> .	7080

AGENT_SERVER_SSL_PORT	Port on server to use for SSL communications with the server. The value is saved to agent.setup.camSSLPort in agent.properties.	7443
-----------------------	---	------

Example Silent Mode MSI Invocations

- To silently install Hyperic Server and the Hyperic Agent on a local machine under "C:\hyperic":

```
%Comspec% /c msixec /i "installer_path\hyperic-hq-installer-4.6.n.build.msi" /qn
INSTALLDIR="C:\hyperic"
```

- To install (locally) a Hyperic Agent that will communicate securely with the Hyperic Server at 69.59.181.106:

```
%Comspec% /c msixec /i "installer_path\hyperic-hq-installer-4.6.n.build.msi" /qn ADDLOCAL=Agent
AGENT_IS_SECURE=1 AGENT_SERVER_ADDRESS=69.59.181.106 AGENT_SERVER_USER=hqadmin
AGENT_SERVER_PASSWORD=password
```

- To install the Hyperic Server and the Hyperic Agent on a local machine using an MSI installer on a remote machine that is accessible on the network:

```
%Comspec% /c msixec /i "\\network_path\hyperic-hq-installer-4.6.n.build.msi" PROPERTY1=VALUE1
PROPERTY2=VALUE2 ...
```

- To silently install the Hyperic Server and the Hyperic Agent to a local machine using an MSI installer on a remote machine that is accessible on the network, include the /qn switch to turn off the user interface:

```
%Comspec% /c msixec /i "\\network_path\hyperic-hq-installer-4.6.n.build.msi" /qn
PROPERTY1=VALUE1PROPERTY2=VALUE2...
```

Silent MSI installation to Multiple Hosts Using Push Techniques

This section describes alternatives for doing silent MSI installs to multiple machines.

Using AT or SOON to start a process on a remote workstation

The AT and SOON commands can be used to schedule commands at a future time. AT, which is built into the command processor, schedules commands and programs to run on a local or remote computer at a specified time. Instead of running processes at a specific time, the SOON command runs them after a specified delay. SOON.EXE is available as a free Microsoft download.

Here are examples of how to run these commands:

```
AT targetPC 10:30 /INTERACTIVE \\myPC\myShare\quietInstall.bat
SOON targetPC 30 /INTERACTIVE \\myPC\myShare\quietInstall.bat
```

Executing processes on a remote system has security implications.

- The local machine must have sufficient privileges to start a batch routine on a remote system.

- You must establish privileges for the remote system to access network resources when running the install batch routine. When the command processor runs your batch routine on the target system, it executes with Local System privileges. It is therefore necessary for the batch routine to open a privilege pipe to the network resource containing the MSI package. A workaround is to add a NET command to your batch routine, as demonstrated in this sample quietInstall.bat:

```
net use * \\myPC\myShare /user:domain\username password /persistent:no
%Comspec% /c msixec /i "\\myPC\myShare\hyperic-hq-installer-4.6.n.build.msi" /qn
```

Using PsExec to Start a Process on a Remote System

The Windows PsExec utility is a free, light-weight telnet replacement you can use to run processes on other systems, complete with full interactivity for console applications, without having to manually install client software. PsExec's most powerful uses include launching interactive command prompts on remote systems and remote-enabling tools to show information about remote systems. It can be downloaded as part of the Sysinternal PsToolspackage.

PsExec can be used to run the batch routine at a remote workstation by invoking the following command:

```
psexec targetPC -u domain\username -p password -i -c -f
\\myPC\myShare\quietInstall.bat
```

In the example above, domain\username has local administrative privileges for the targetPC machine. In addition, it should have the necessary privileges to access the myShare folder on the myPC machine.

The batch file quietInstall.bat is used to invoke the MSI installer over the network on the myPC machine. A sample quietInstall.bat might contain the following command:

```
%Comspec% /c msixec /i "\\myPC\myShare\Hyperic HQ 4.6.msi" /qn
```

Remote Installs with Microsoft Management Console

This section has information on how to automatically install components to a group of machines, using Microsoft Management Control and Active Directory.

With Windows Group Policy, Hyperic components can be automatically installed on a group of machines by performing the following steps:

1. Log on to the domain controller.
2. Copy the MSI file into a folder that is shared with access granted to all target machines.
3. Open the Microsoft Management Control (MMC) Active Directory Users and Computers snap-in.
4. Navigate to the group of computers onto which a Hyperic component is to be deployed.
5. Open Properties.
6. Open Group Policies.
7. Add a new policies, and edit it.
8. In Computer Configuration/Software Installation, chose New/Package.
9. Select the MSI file through the network path.
10. Optionally, select that you want Hyperic to be uninstalled if the computer leaves the scope of the policy.

Group policy propagation typically takes some time. In order to reliably deploy the Hyperic MSI package, all machines should be rebooted.

Solving Service Startup Problems After MSI Install

If you install the Hyperic Server and the Hyperic Agent on the same machine, and accept the default "Start Hyperic HQ Services when install completes" option, agent startup problems can result.

In this scenario, as a last step, the installer will issue a server start command, followed by an agent start command.

The agent must contact the server to start up successfully. If the machine the Hyperic components run on is slow or busy, the Hyperic Server can take a long time to start. The Hyperic Agent makes a finite number of attempts to connect to the server, and if it continues to fail, the agent gets stuck. No software will be auto-discovered on the platform and the agent will not appear in the Hyperic user interface.

To solve this problem, force the agent to repeat the setup process by entering this command in a shell:

```
AGENT_HOME/bin/hq-agent.bat setup
```

Install Hyperic Server RPM

- [About Hyperic Server RPM \(see page 72\)](#)
- [Step 1 - Install vFabric License Server \(see page 73\)](#)
- [Step 2 - Activate vFabric License \(see page 73\)](#)
- [Step 3 - Install vFabric Repository RPM \(see page 73\)](#)
 - [Instructions for 5.0 and 5.1 Repositories \(see page 73\)](#)
 - [Instructions for 5.2 Repository \(see page 74\)](#)
- [Step 4 - Specify Server Configuration \(see page 75\)](#)
- [Step 5 - Install Hyperic Server \(see page 76\)](#)
- [Step 6 - Start the Hyperic Server \(see page 76\)](#)
- [After Installing Hyperic Server \(see page 77\)](#)
- [Server Properties File for RPM Installation \(see page 77\)](#)

About this page...

This page has instructions for installing Hyperic Server 4.6 from an RPM to an RHEL 5 or RHEL 6 virtual machine

Before running the installer, see [Hyperic Installation and Startup Process \(see page 20\)](#) for information about where server installation fits in the implementation process. For production environments, there are several steps to perform before installing the Hyperic Server, including database setup and SSL configuration.



Installing a downloaded RPM?

Steps 1 through 3 apply if you are installing Hyperic Server from the VMware RPM repository. If instead, you have downloaded the server RPM, skip to [Step 4 - Specify Server Configuration \(see page 75\)](#).

About Hyperic Server RPM

The name of the vFabric Hyperic Server RPM package is `vfabric-hyperic-server`, appended by an architecture suffix, such as `.noarch` or `.x86_64`.

When you install the server RPM, there are two ways to define the required installation options:

- You can specify required Hyperic Server installation options in a properties file prior to installing the server, as described in [Step 4 - Specify Server Configuration in Properties File \(see page 75\)](#), or
- You can supply configuration options after installing the server, using one of the methods described in [Configure Hyperic Server After RPM Installation \(see page \)](#).

The installation owner and the installation group are both `hyperic`

By default, yum will install the server in `/opt/hyperic/server-current`, where `/opt/hyperic/server-current` is a symbolic link to a sibling versioned directory, such as `/opt/hyperic/server-4.6-EE`.

Step 1 - Install vFabric License Server

Install the vFabric License Server on the VM where your existing vCenter Server is installed. For instructions, see "[Install vFabric License Server on Your vCenter Server \(http://pubs.vmware.com/vfabric51/topic/com.vmware.vfabric.platform.5.1/vfabric/install.html\)](#)" in *vFabric Suite 5.1*.

Step 2 - Activate vFabric License

Active your vFabric Hyperic license. For instructions, see "[Activate vFabric Platform Licenses \(http://pubs.vmware.com/vfabric51/topic/com.vmware.vfabric.platform.5.1/vfabric/install.html\)](#)" in *vFabric Suite 5.1*.

Step 3 - Install vFabric Repository RPM

In this step you install the vFabric Repository RPM on the RHEL VM where you will install the Hyperic Server. This RPM allows you to browse the vFabric repository.

vFabric RPM repositories are located on `repo.vmware.com`.

Following the instructions for you version of vFabric Suite:

- [Instructions for 5.0 and 5.1 Repositories \(see page 73\)](#)
- [Instructions for 5.2 Repository \(see page 74\)](#)

Instructions for 5.0 and 5.1 Repositories

- `vfabric-5-2-repo` — vFabric component RPMs that are certified for vFabric Platform 5.
- `vfabric-5.1-repo-5.1-1` — vFabric component RPMs that are certified for vFabric Suite 5.1.
- `vfabric-all-repo` — vFabric component RPMs for customers who have not bought a vFabric 5 Standard or Advanced license, and more recently released versions of vFabric components that may not necessarily be certified to work with those of vFabric 5 or vFabric 5.1. (Note: This repository is deprecated in vFabric Suite 5.2)

If you have a vFabric Standard or Advanced license, plan to install only RPMs that are certified for vFabric 5 or 5.1, then install `vfabric-5-repo-5-2` or `vfabric-5.1-repo-5.1-1`, as appropriate. If you do not have a vFabric Standard or Advanced license, or want to install a maintenance release that was not certified for vFabric 5 or vFabric 5.1, install `vfabric-all-repo`.

To install the 5.0, 5.1, or the `vfabric-all-repo` repository:

1. Log in to the RHEL VM as the `root` user (or as a user with `sudo` privileges).
2. Open a terminal window.

3. Install the desired repository RPMs. Note that the URLs of the RPMs vary depending on whether the version of RHEL you use.

- For RHEL 5:

```
rpm -Uvh http://repo.vmware.com/pub/rhel5/vfabric/5/vfabric-5-repo-5-2.noarch.rpm
\\ to install the vFabric 5 repository RPM
rpm -Uvh http://repo.vmware.com/pub/rhel5/vfabric/5.1/vfabric-5.1-repo-5.1-1.noarch.rpm
\\ to install the vFabric 5.1 repository RPM
rpm -Uvh http://repo.vmware.com/pub/rhel5/vfabric-all/vfabric-all-repo-1-1.noarch.rpm
\\ to install the "all-repo" repository RPM
```

- For RHEL 6:

```
rpm -Uvh http://repo.vmware.com/pub/rhel6/vfabric/5.1/vfabric-5.1-repo-5.1-1.noarch.rpm \\
to install the vFabric 5 repository RPM
rpm -Uvh http://repo.vmware.com/pub/rhel6/vfabric-all/vfabric-all-repo-1-1.noarch.rpm \\
to install the "all-repo" repository RPM
```

 Use `sudo` to run the preceding commands if you are not logged in as root.

Searching the RPM repository

You can use the `yum search vfabric` command to list of vFabric components that you can install from the VMware repository. For example:

```
yum search vfabric
...
===== Matched: vfabric
=====
vfabric-eula.noarch : VMware vFabric 5.1 End User License Agreement
vfabric-gemfire.noarch : VMware vFabric GemFire
vfabric-hyperic-agent.noarch : VMware vFabric Hyperic Agent
vfabric-hyperic-server.x86_64 : VMware vFabric Hyperic Server
vfabric-insight-agent.noarch : com.springsource.insight.dist.vfabric:agent-rpm
vfabric-insight-dashboard.noarch : com.springsource.insight.dist.vfabric:dashboard-rpm
vfabric-rabbitmq-java-client-bin.noarch : The RabbitMQ Java Client Library
vfabric-rabbitmq-server.x86_64 : The RabbitMQ server
vfabric-sqlfire.noarch : VMware vFabric SQLFire
vfabric-tc-server-standard.noarch : VMware vFabric tc Server Standard
vfabric-web-server.x86_64 : VMware vFabric Web Server
```

Instructions for 5.2 Repository

To install the vFabric 5.2 repository:

1. On the RHEL computer, start a terminal either as the root user or as an unprivileged user who has `sudo` privileges.

2. Install the vFabric repository RPM using the following `wget` command, passing it the appropriate URL. The URL differs depending on the version of RHEL you are using.

- Note: Run the entire `wget` command on a single line. Include the `| sh` at the end, or the RPM installation will not work.
- For RHEL 5:

```
wget -q -O - http://repo.vmware.com/pub/rhel5/vfabric/5.2/vfabric-5.2-suite-installer | sh
```

- For RHEL 6:

```
wget -q -O - http://repo.vmware.com/pub/rhel6/vfabric/5.2/vfabric-5.2-suite-installer | sh
```

- If necessary, use `sudo` to run the preceding commands if you are not logged in as the root user. For example:

```
sudo wget -q -O - http://repo.vmware.com/pub/rhel6/vfabric/5.2/vfabric-5.2-suite-installer | sh
```

The command performs the following tasks:

1. Imports the vFabric GNU Privacy Guard (GPG) key.
2. Installs the vFabric 5.2 repository RPM.
3. Launches the VMware End User License Agreement (EULA) acceptance and repository configuration script.
4. Outputs the EULA for you to read; you must answer yes to accept the terms and continue.

Step 4 - Specify Server Configuration


This section has instructions for configuring Hyperic Server in a properties file prior to installation. If you do not create a properties file prior to installing the server, the installer will configure the Hyperic Server in accordance with a default properties file.

To define Hyperic Server installation options in a properties file:

1. Running as `root`, or using `sudo`, create a directory for the properties file:

```
mkdir -p /etc/vmware/vfabric/hyperic
```


2. Specify the server installation settings in a file named `vfabric_hyperic_server.properties`.

 You must name the file exactly as described and put it in the specified location for the RPM installation to work correctly.

- See [Server Properties File for RPM Installation](#) (see page 77) for a listing of a sample `vfabric_hyperic_server.properties` file. Note that the example properties file configures Hyperic Server to use the built-in local PostgreSQL database. You can edit the properties file to configure Hyperic Server to use an external — either local or remote — PostgreSQL, MySQL, or Oracle database.
3. Copy `vfabric_hyperic_server.properties` to the `/etc/vmware/vfabric/hyperic/` directory.

4. Install Hyperic Server following the instructions in the next section.

Step 5 - Install Hyperic Server

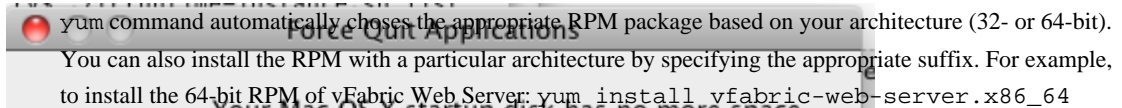
-  Start local SMTP server on Hyperic Server host

The RPM installer requires that your SMTP server is listening on port 25 on the Hyperic Server host.

To install Hyperic Server:

1. On the platform where you wish to install the Hyperic Server:

```
yum install vfabric-hyperic-server
```



yum command automatically chooses the appropriate RPM package based on your architecture (32- or 64-bit). You can also install the RPM with a particular architecture by specifying the appropriate suffix. For example, to install the 64-bit RPM of vFabric Web Server: `yum install vfabric-web-server.x86_64`

2. yum resolves dependencies and displays the packages it plans to install.




If this is the first time that you have installed a vFabric component on the VM, yum also installs the `vfabric-eula` RPM and prompts you to accept the VMware license agreement.

3. Enter `y` at the prompt to start the installation.
4. What happens next depends on whether you specified the server configuration in the `/etc/vmware/vfabric/hyperic/vfabric_hyperic_server.properties` file.
 - If you *did* create the properties file, the setup dialog and responses from the properties file are displayed, and an "Installation Complete" message appear. After seeing this message, proceed to the [Start the Hyperic Server \(see page 23\)](#) step of [Hyperic Installation and Startup Process \(see page 20\)](#).
 - If you *did not* specify the server configuration in the properties file, the following message is displayed:

```
/etc/vmware/vfabric/hyperic/vfabric_hyperic_server.properties, not found. HQ will be installed, but not configured.  
  
Please use /opt/hyperic/hyperic-hqee-installer/setup.sh to interactively configure HQ.
```

Step 6 - Start the Hyperic Server

-  Need to configure a license?

If you obtained Hyperic Server standalone, rather than as a part of vFabric Suite, perform the steps in [Install or Configure vFabric Hyperic License \(see page 114\)](#) before starting the Hyperic Server.

yum installs Hyperic Server to start automatically each time your RHEL VM is booted. The init script is `/etc/init.d/hyperic-hq-server`.

Follow these steps to start the Hyperic server manually:

1. Log in to the RHEL VM as `root`.
2. Open a terminal window and run the init script:

```
/etc/init.d/hyperic-hq-server start
```

The first time Hyperic Server starts it may take a few minutes to initialize; subsequent startups will be faster.

3. After Hyperic Server starts, invoke the Hyperic user interface in your browser:

```
http://host:7080
```

- Where `host` is the hostname of the RHEL VM on which you just installed Hyperic Server. `##` If your browser is on the same computer on which you installed Hyperic Server, you can use `localhost` rather than the name of the computer.
4. Log in using the default administration username/password of `hqadmin/hqadmin`; be sure to change the password after you log into the Hyperic UI.

After Installing Hyperic Server

- For information about installing the Hyperic Agent, see [Install Hyperic Agent RPM \(see page 79\)](#).
- For information and required and optional Hyperic Server Configuration see [Hyperic Installation and Startup Process \(see page 20\)](#).

Server Properties File for RPM Installation

This section contains a listing of `vfabric_hyperic_server.properties`.

```
# Properties file for vFabric Hyperic Server Configuration
#
# This file must be place in /etc/vmware/vfabric/hyperic/ with a name of
# vfabric_hyperic_server.properties to be used by the vfabric-hyperic-server
# rpm for vFabric Hyperic Server configuration.
#
#
# To configure the build-in local Postgresql database uncomment the below
# sections. See below for other database types.
#
#####
# Configuration of local built-in Postgresql database
#
# Use the local built-in Postgresql database instead of other database types
BUILT_IN_POSTGRESQL=yes
#
# Do you accept the terms of the agreement?
HQ_ACCEPT_EULA=y
```

```

# HQ server installation path
HQ_SERVER_INSTALL_PATH=/opt/hyperic

# email address that HQ will use as the sender for email messages
HQ_SENDER_EMAIL_ADDRESS=hqadmin@eng.vmware.com

# encryption key to use to encrypt the database password
HQ_DB_CRYPT_KEY=12345678

# username be for the initial admin user
HQ_ADMIN_USER=hqadmin

# password be for the initial admin user
HQ_ADMIN_PASSWORD=hqadmin

# email address be for the initial admin user
HQ_ADMIN_EMAIL_ADDRESS=hqadmin@eng.vmware.com

# End of configuration for local built-in Postgresql database
#####

# To configure HQ with a local or remote database other than the built-in
# local instance of Postgresql comment out the above section and uncomment
# the properties in the section below. Support databases include local or remote
# versions of MySQL, Oracle, and Postgresql.
#
#####
# For configuration with local or remote MySQL, Oracle, and Postgresql databases

### Do you accept the terms of the agreement?
#HQ_ACCEPT_EULA=y

### HQ server installation path
#HQ_SERVER_INSTALL_PATH=/opt/hyperic

### email address that HQ will use as the sender for email messages
#HQ_SENDER_EMAIL_ADDRESS=hqadmin@eng.vmware.com

### database type of [mysql|oracle|postgesql]
#HQ_DB_TYPE=mysql

### JDBC connection URL for HQ database
#HQ_JDBC=jdbc:mysql://10.0.0.1:3306/hqdb

### username to use to connect to the database
#HQ_DB_USERNAME=hqadmin

### password to use to connect to the database
#HQ_DB_PASSWORD=hqadmin

### encryption key to use to encrypt the database password
#HQ_DB_CRYPT_KEY=12345678

### username be for the initial admin user
#HQ_ADMIN_USER=hqadmin

### password be for the initial admin user
#HQ_ADMIN_PASSWORD=hqadmin

### email address be for the initial admin user

```

```
#HQ_ADMIN_EMAIL_ADDRESS=hqadmin@eng.vmware.com

# End of configuration
#####
```

Install Hyperic Agent RPM

- [About Hyperic Agent RPM \(see page 79\)](#)
- [Step 1 - Install vFabric Repository RPM \(see page 79\)](#)
 - [Instructions for 5.0 and 5.1 Repositories \(see page 80\)](#)
 - [Instructions for 5.2 Repository \(see page 81\)](#)
- [Step 2 - Install Agent RPM \(see page 82\)](#)
- [Step 3 - Configure and Start Hyperic Agent \(see page 82\)](#)

About this page...

This page has instructions for installing Hyperic Agent 4.6 from an RPM from the VMware YUM repository.

Before running the installer, see [Hyperic Installation and Startup Process \(see page 20\)](#) for information about where agent installation fits in the implementation process.

About Hyperic Agent RPM

The name of the vFabric Hyperic Agent RPM package is `vfabric-hyperic-agent.noarch`. Note that the agent in the RPM does not include a JRE. Agent hosts must have the J2RE virtual package installed. A Sun 1.6 JRE is required. The RPM:

- Creates the user and group named "hyperic" if they do not exist.
- Sets the home directory of the hyperic user to `/opt/hyperic`.
- Installs the agent files into `/opt/hyperic/hyperic-hqee-agent`.
- Installs an init script to `/etc/init.d/hyperic-hqee-agent`.
- Adds init script to `chkconfig` and sets it to "on" for runlevels 2, 3, 4, and 5.

-
- ▶ Open firewall port on Hyperic Agent, if necessary

If `iptables` (a host-based firewall tool typically enabled by default on Redhat and Fedora installations) is configured, you may need to open up the Hyperic Agent's listen port for communication from the Hyperic Server, using a command similar to:

```
~/sbin/iptables -ARH-Firewall-1-INPUT -p tcp --dport 2144 -j ACCEPT`
```

If SELinux is enabled, additional configuration may be required.

Step 1 - Install vFabric Repository RPM

In this step you install the vFabric Repository RPM on the RHEL VM where you will install the Hyperic Server. This RPM allows you to browse the vFabric repository.

vFabric RPM repositories are located on `repo.vmware.com`.

Following the instructions for your version of vFabric Suite:

- [Instructions for 5.0 and 5.1 Repositories \(see page 80\)](#)
- [Instructions for 5.2 Repository \(see page 81\)](#)

Instructions for 5.0 and 5.1 Repositories

- `vfabric-5-2-repo` — vFabric component RPMs that are certified for vFabric Platform 5.
- `vfabric-5.1-repo-5.1-1` — vFabric component RPMs that are certified for vFabric Suite 5.1.
- `vfabric-all-repo` — vFabric component RPMs for customers who have not bought a vFabric 5 Standard or Advanced license, and more recently released versions of vFabric components that may not necessarily be certified to work with those of vFabric 5 or vFabric 5.1. (Note: This repository is deprecated in vFabric Suite 5.2)

If you have a vFabric Standard or Advanced license, plan to install only RPMs that are certified for vFabric 5 or 5.1, then install `vfabric-5-repo-5-2` or `vfabric-5.1-repo-5.1-1`, as appropriate. If you do not have a vFabric Standard or Advanced license, or want to install a maintenance release that was not certified for vFabric 5 or vFabric 5.1, install `vfabric-all-repo`.

To install the 5.0, 5.1, or the `vfabric-all-repo` repository:

1. Log in to the RHEL VM as the `root` user (or as a user with `sudo` privileges).
2. Open a terminal window.
3. Install the desired repository RPMs. Note that the URLs of the RPMs vary depending on whether the version of RHEL you use.
 - For RHEL 5:

```
rpm -Uvh http://repo.vmware.com/pub/rhel5/vfabric/5/vfabric-5-repo-5-2.noarch.rpm
\\ to install the vFabric 5 repository RPM
rpm -Uvh http://repo.vmware.com/pub/rhel5/vfabric/5.1/vfabric-5.1-repo-5.1-1.noarch.rpm
\\ to install the vFabric 5.1 repository RPM
rpm -Uvh http://repo.vmware.com/pub/rhel5/vfabric-all/vfabric-all-repo-1-1.noarch.rpm
\\ to install the "all-repo" repository RPM
```

- For RHEL 6:

```
rpm -Uvh http://repo.vmware.com/pub/rhel6/vfabric/5.1/vfabric-5.1-repo-5.1-1.noarch.rpm \\
to install the vFabric 5 repository RPM
rpm -Uvh http://repo.vmware.com/pub/rhel6/vfabric-all/vfabric-all-repo-1-1.noarch.rpm \\
to install the "all-repo" repository RPM
```

 Use `sudo` to run the preceding commands if you are not logged in as `root`.

Searching the RPM repository

You can use the `yum search vfabric` command to list of vFabric components that you can install from the VMware repository. For example:

```
yum search vfabric
...
===== Matched: vfabric
=====
vfabric-eula.noarch : VMware vFabric 5.1 End User License Agreement
vfabric-gemfire.noarch : VMware vFabric GemFire
vfabric-hyperic-agent.noarch : VMware vFabric Hyperic Agent
vfabric-hyperic-server.x86_64 : VMware vFabric Hyperic Server
vfabric-insight-agent.noarch : com.springsource.insight.dist.vfabric:agent-rpm
vfabric-insight-dashboard.noarch : com.springsource.insight.dist.vfabric:dashboard-rpm
vfabric-rabbitmq-java-client-bin.noarch : The RabbitMQ Java Client Library
vfabric-rabbitmq-server.x86_64 : The RabbitMQ server
vfabric-sqlfire.noarch : VMware vFabric SQLFire
vfabric-tc-server-standard.noarch : VMware vFabric tc Server Standard
vfabric-web-server.x86_64 : VMware vFabric Web Server
```

Instructions for 5.2 Repository

To install the vFabric 5.2 repository:

1. On the RHEL computer, start a terminal either as the root user or as an unprivileged user who has sudo privileges.
2. Install the vFabric repository RPM using the following `wget` command, passing it the appropriate URL. The URL differs depending on the version of RHEL you are using.
 - Note: Run the entire `wget` command on a single line. Include the `| sh` at the end, or the RPM installation will not work.
 - For RHEL 5:

```
wget -q -O - - http://repo.vmware.com/pub/rhel5/vfabric/5.2/vfabric-5.2-suite-installer | sh
```

- For RHEL 6:

```
wget -q -O - - http://repo.vmware.com/pub/rhel6/vfabric/5.2/vfabric-5.2-suite-installer | sh
```

- If necessary, use `sudo` to run the preceding commands if you are not logged in as the root user. For example:

```
sudo wget -q -O - - http://repo.vmware.com/pub/rhel6/vfabric/5.2/vfabric-5.2-suite-installer
| sh
```

The command performs the following tasks:

1. Imports the vFabric GNU Privacy Guard (GPG) key.
2. Installs the vFabric 5.2 repository RPM.
3. Launches the VMware End User License Agreement (EULA) acceptance and repository configuration script.
4. Outputs the EULA for you to read; you must answer yes to accept the terms and continue.

Step 2 - Install Agent RPM

To install and configure the Hyperic Agent from a downloaded RPM:

1. On the platform the agent will manage, use yum to install the agent

```
yum install vfabric-hyperic-agent
```

2. If you have not already done so, install a JDK or JRE on the platform.
3. As the root user, edit the `/etc/init.d/hyperic-hqee-agent` file, setting the `HQ_JAVA_HOME` to home directory of the JDK or JRE you want the agent to use.

Step 3 - Configure and Start Hyperic Agent

Follow the instructions for configuring and starting the agent, starting with the [Configure Hyperic Agent in properties file](#) (see page 24) step of [Hyperic Installation and Startup Process](#) (see page 20).

Set Up Agent Interactively

- [About the Agent Configuration Dialog](#) (see page 82)
- [Launch the Configuration Dialog](#) (see page 83)
- [Agent Configuration Dialog](#) (see page 83)

About this page...

This page has instructions for interactively configuring a newly installed Hyperic Agent to communicate with the Hyperic Server, or reconfiguring the communication behavior of an existing agent.

Before following the instructions below, see [Hyperic Installation and Startup Process](#) (see page 20) to see how agent configuration fits into the overall Hyperic implementation process and to understand agent configuration options and requirements.

Depending upon your situation, it may be preferable to configure agent-server communication settings in the agent's `agent.properties` file, rather than interactively:

- *Some communication features, such as user-managed SSL keystore location, can only be configured in an agent's properties file. For more information, see the [Configure Hyperic Agent in properties file](#) (see page 24) step of [Hyperic Installation and Startup Process](#) (see page 20).*
- *If you have a lot of agents to configure, properties file configuration is faster than interactive configuration, as described in [Install Hyperic Agents in Volume](#) (see page 100).*

About the Agent Configuration Dialog

The agent configuration dialog appears in the shell when you launch a Hyperic Agent that lacks the configuration values that specify the location of the Hyperic Server. The dialog queries for the address and port of the Hyperic server, and other connection-related data. To understand the details of how the agent figures out where to contact the Hyperic Server see [What Happens When an Agent Starts Up](#) (see page 13) on [About the Agent Launcher and Agent Startup](#) (see page 13).

The agent configuration dialog is presented in these cases:

- The first time you start an agent (assuming that have not supplied the properties in `agent.profile`).
- When you start an agent that whose saved server connection data is corrupt or has been removed.
- When you run the agent launcher with the `setup` option, which causes the agent to prompt for new connection property settings.

You can also run the agent launcher with the to re-run the configuration dialog

Launch the Configuration Dialog

1. Open a terminal window on the platform where the agent is installed.
2. Navigate to the `AgentHome/bin` directory.
3. Run the agent launcher with the `start` or `setup` option:

- On Unix-like platforms:

```
sh hq-agent.sh start
```

- On Windows platforms, install the Windows service for the agent, and then start it:


```
sh hq-agent.bat install
sh hq-agent.bat start
```

4. Respond to the prompts described below in [Agent Configuration Dialog \(see page 83\)](#).

Agent Configuration Dialog

1. *Should Agent communications to HQ be unidirectional [default=no]*
 - This prompt only appears if you are installing vFabric Hyperic. To understand this option, see [About Agent - Server Communication \(see page 15\)](#).
2. *What is the HQ server IP address*
 - Enter the listen address of your Hyperic Server. The server must be running. If the server is on the same machine as the agent, you can enter `localhost`.
 - If there is a firewall blocking traffic from the agent to the server, specify the address of the firewall.
3. *Should Agent communications to HQ always be secure [default=yes].*
 - Accept the default to configure Hyperic to use SSL for agent-to-server communication. (Server-agent communication is *always* SSL.)
 - SSL configuration is strongly recommended, and **required** if you configured unidirectional communications in Step 1.
 - To configure Hyperic to use plain HTTP for agent-to-server communication, enter "no".
4. You are prompted to identify the server port where the Hyperic server listens for agent communication. Depending on whether you selected SSL in the previous step, one of the following prompts appears:
 - *What is the secure HQ server port [default=7443]* — If the Hyperic Server is configured to listen for SSL communications on the default SSL port of 7443, press **Return**.
 - *What is the HQ server port [default=7080]* — If the Hyperic Server is configured to listen on the default port of 7080, press **Return**. If there is a firewall blocking traffic from the agent to the server, configure it to forward traffic on TCP port 7080 (or 7443) to the host running the Hyperic Server.
5. *What is your HQ login [default=hqadmin]:*
 - By default, the Hyperic Server is initially configured with an administrative account with username `hqadmin`. Unless you have configured a different Hyperic user account for agent-server communications, accept the default.

6. *What is your HQ password*
 - Enter the password for the username you supplied at the previous prompt.
7. *What IP should HQ use to contact the agent [default=n.n.n.n]*
 - The prompt will show the first IP address the agent detects on the host.
 - If there is another IP address on the host you prefer to use, enter it.
 - If there is a firewall blocking traffic from the server to the agent, enter the IP address of the firewall, and configure the firewall to forward traffic intended for the Hyperic Agent to the listen address of the agent host.
8. *What port should HQ use to contact the agent [default=2144]*
 - Enter the agent port the Hyperic Server should use when it initiates contact with the agent. Specify the port that the agent binds to at startup, which by default is 2144.
 - If you have previously edited `agent.properties` to explicitly define a different listen port, using the optional `agent.listenPort` property, that is the value you should supply to this prompt.
 - If there is a firewall blocking traffic from the server to the agent, configure the device to forward traffic on TCP port 2144 to the Hyperic Agent.
9. After you respond to the last prompt the agent initiates a connection to the server and the server verifies that it can communicate with the agent.

 If you have not configured user-managed keystores on the agent and server the agent will issue warning related to the components' self-signed SSL certificates. For more information see the [SSL Warnings](#) (see page 20) note on [Hyperic Installation and Startup Process](#) (see page 20).

- Once the agent and server successfully establish communication and complete the registration process, messages similar to the following appear:

```
Received temporary auth token from agent
Registering agent with HQ
HQ gave us the following agent token
1215038691323-8570363106994871928-8259195015465958356
Informing agent of new HQ server
Validating
Successfully setup agent
```

10. The Hyperic Agent discovers the platform and supported products running on it. For more information, see the [Import discovered resources](#) (see page 26) step on [Hyperic Installation and Startup Process](#) (see page 20).

Set Up Agent in Properties File

*Topics marked with * relate to features available only in vFabric Hyperic.*

- [Agent Properties Location](#) (see page 85)
- [Procedure: Configure Agent-Server Communication Properties](#) (see page 85)
 - [Step 1 - Open or create agent.properties](#) (see page 85)
 - [Step 2 - Uncomment Agent-Server Communication Properties](#) (see page 85)
 - [Step 3 - Define Communication Properties in agent.properties File](#) (see page 86)
 - [Step 4 - Configure Unidirectional Communications \(Optional\)](#) (see page 87)
 - [Step 5 - Configure Agent Keystore \(Optional\)](#) (see page 87)
 - [Step 6 - Configure Additional Agent Behaviors \(Optional\)](#) (see page 87)
 - [Step 7 - Save Changes](#) (see page 88)
- [Encrypt an Agent Property Value](#) (see page 88)
- [Communication Properties Reference](#) (see page 89)

About this page...

This page has instructions for configuring a newly installed Hyperic Agent to communicate with the Hyperic Server in its properties file. It corresponds to the [Configure Hyperic Agent in properties file \(see page 24\)](#) step of the [Hyperic Installation and Startup Process \(see page 20\)](#).

Note that if you start an agent without previously configuring required data in the properties file, you will be prompted to supply connection-related data in the shell, as described in [Set Up Agent Interactively \(see page 82\)](#).

If you have multiple Hyperic Agents to deploy, it is more efficient to configure agent behaviors in the properties files. [Install Hyperic Agents in Volume \(see page 100\)](#) describes options for remote agent installation and configuration using a standard `agent.properties` file that you can use for all agents in your environment.

Agent Properties Location

The Hyperic Agent looks for its properties file in two locations, in this order:

- `HqUserHome/.hq` — If this directory exists and contains `agent.properties`, the Hyperic Agent will use the property values defined there. (Hyperic honors this location for historical reasons. In pre-4.2 versions of Hyperic, storing the properties file in a location external to the agent installation directory was a method of ensuring that agent configuration settings survived an agent upgrade. This precaution is not necessary in Hyperic 4.2 — you can update the agent bundle from the Hyperic user interface without risk of overwriting the properties settings.)
- `AgentHome/conf` — This is the default location of `agent.properties`.

If the agent does not find the the properties it needs to establish communications with the Hyperic Server in either of these locations, it prompts for the property values at startup.

Procedure: Configure Agent-Server Communication Properties

Step 1 - Open or create `agent.properties`

Make a copy of the `agent.properties` file from the agent installation.

Step 2 - Uncomment Agent-Server Communication Properties

In the `agent.properties` file, find the section excerpted below, and remove the hash mark (#) in front of each the properties shown at the end of the excerpt.

```

## Use the following if you'd like to have the agent setup
## automatically from these properties. The values for these
## properties are used to answer the setup questions
##
## If any of these properties are left undefined, the setup
## process will prompt for their values
##
## If the value that should be used is the default when interactive
## setup is done, use the string *default* as the value for the option

#agent.setup.camIP=localhost
#agent.setup.camPort=7080
#agent.setup.camSSLPort=7443
#agent.setup.camSecure=yes
#agent.setup.camLogin=hqadmin
#agent.setup.camPword=hqadmin
#agent.setup.agentIP=*default*
#agent.setup.agentPort=*default*
#agent.setup.resetupTokens=no

```

Step 3 - Define Communication Properties in agent.properties File

See [Communication Properties Reference \(see page 89\)](#) below for property definitions.

The `agent.properties` file contains properties you can configure to govern both agent-initiated and server-initiated communication.

- Specify the location and credentials the agent should use to contact the Hyperic Server with these properties:
 - `agent.setup.camIP` — Specify the address or hostname of the Hyperic Server.
 - `agent.setup.camPort` — The default value is the standard plaintext Hyperic Server listen port. If your server is configured for a different listen port, supply the port number.
 - `agent.setup.camSSLPort` — The default value is the standard SSL Hyperic Server listen port. If your server is configured for a different listen port, supply the port number.
 - `agent.setup.camSecure` — The default value is "yes" (use SSL). SSL configuration is strongly recommended, and **required** if you are going to configure the agent for unidirectional communications. Change to "no" if you do not require the agent to use secure communications when contacting the Hyperic Server.
 - `agent.setup.camLogin` — Specify the username the agent should use when connecting to the server. If you change the value from the default value ("hqadmin"), make sure that that user account is properly configured on the Hyperic Server.
 - `agent.setup.camPword` — Specify the password the agent should use, along with the username above, when connecting to the server. Make sure that the password is the one configured in Hyperic for the user account.

About Password Encryption

The first time you start the Hyperic Agent, if `agent.setup.camPword` is uncommented, and has a plain text value, the agent will encrypt the value. Note that you can encrypt this and other agent property values yourself, as described in [Encrypt an Agent Property Value \(see page 88\)](#).

- You may specify the address or hostname and the listen port the Hyperic Server should use to contact the Hyperic Agent with the following properties. Note however, that if you are creating a standard `agent.properties` file that can be used for all agents (as described in [Install Hyperic Agents in Volume \(see page 100\)](#)), uncomment these properties, but *do not* change the value of either:
 - `agent.setup.agentIP` — If you leave the default setting — "**default**" — the Hyperic Agent will detect an IP address on the platform and choose it as its listen address.
 - `agent.setup.agentPort` — If you leave the default setting — "**default**" — the Hyperic Agent will use the default listen port (2144 for plaintext or 2443 for SSL) as its listen address. If that port is unavailable, the agent will detect a free port and choose it as its listen port.

These are the minimum properties required for agent-server communication.

Step 4 - Configure Unidirectional Communications (Optional)

By default, agent-server communication is bidirectional. If your policies dictate that all communication between agent and server are agent-initiated, you can uncomment the `agent.setup.unidirectional` property and set it to "yes". For more information, see [Unidirectional Agent-Server Communication \(see page 16\)](#).

If you configure unidirectional communication, and the agent will contact the Hyperic Server via a proxy server, define the proxy server host in the following properties, which you must add to the properties file:

- `agent.proxyHost`
- `agent.proxyPort`

See [agent.setup.unidirectional \(see page 93\)](#), [agent.proxyHost \(see page 97\)](#) and [agent.proxyPort \(see page 98\)](#) below for property definitions.

Step 5 - Configure Agent Keystore (Optional)

Perform this step if you want the agent to use a keystore you configure, rather than have it generate and use a self-signed certificate for SSL communication with the Hyperic Server.

Hyperic and SSL Communication

Read [About SSL in Hyperic \(see page 9\)](#) for information about SSL certificates and certificate verification in Hyperic 4.6 and later.

1. Uncomment the following properties in `agent.properties`. Define the full path to the keystore with `agent.keystore.path` and the keystore password with `agent.keystore.password`. For more information, see [agent.keystore.path \(see page 94\)](#) and [agent.keystore.password \(see page 94\)](#) below.

```
# agent.keystore.path=
# agent.keystore.password=
```

2. If you configured the agent for unidirectional communication as described in step 4 above, add `[agent.keystore.alias]` to the properties file, and set it to the alias for the keystore's primary certificate/private key entry.
3. Verify that `agent.setup.acceptUnverifiedCertificate` is "false".

See [Communication Properties Reference \(see page 89\)](#) below for property definitions.

Step 6 - Configure Additional Agent Behaviors (Optional)

As desired, you can configure additional agent behaviors in the `agent.properties` file. For information about configurable agent behaviors, see:

- Configure Auto-Discovery Scanning and Reporting
- Configure Plugin Loading
- Configure Agent Logging
- Tweak the Agent to Enable a Resource Plugin

Step 7 - Save Changes

After completing your edits, save your changes and start the agent.

For instruction on how to start the agent for the first time, see [Start the Hyperic Agent for the first time \(see page 25\)](#) on the [Hyperic Installation and Startup Process \(see page 20\)](#) page.

Encrypt an Agent Property Value

Starting in Hyperic 4.6.6, the `agent.properties` file supports encrypted property values.

If, prior to first agent startup, you uncomment and assign a plain text value to `agent.setup.camPword` or `agent.keystore.password`, the agent will automatically encrypt the property value, as described in [Hyperic Security Features and Recommendations \(see page 7\)](#).

If you prefer, you can encrypt these (and other, if desired) property values yourself.

About Where the Agent Finds Server Connection Data

Note that upon first successful connection to the Hyperic Server, a Hyperic Agent saves the credentials it used in its `/data` directory. Upon each restart, the agent looks first in that directory for server connection details. Hence, edits to the username and password (`agent.setup.cam.Login` and `agent.setup.camPword`) configured in `agent.properties` have no effect, if the agent has valid connection data its `/data` directory.

To add an encrypted entry to `agent.properties`, run the agent start script (`AgentHome/bin/hq-agent.sh` or `AgentHome/bin/hq-agent.bat`) with the new `set-property` option, and supply the name of the property and the value you wish to encrypt.

Do not use set-property option on an agent upgraded to v4.6.6

The `set-property` option is only supported for newly installed agents. You cannot manually encrypt properties for an agent that you upgraded to 4.6.6 by pushing the 4.6.6 bundle from the Hyperic Server. Note however that if an upgraded agent's `agent.properties` file contains uncommented password properties with plaintext values, they will be automatically encrypted.

The command syntax is:

```
./hq-agent.sh set-property PropertyKey PropertyValue
```


For example, to set the `agent.setup.camPword` to "hqadmin":

```
./bin/ hq-agent.sh set-property agent.setup.camPword hqadmin
```

If the properties file does not already define the property, the property definition is added at the end of the `agent.properties` file; the encrypted value (not plain text) is shown. For example:

```
agent.setup.camPword=ENC(gaSh3I8gg1o1L1EDHHJo/g==)
```

The key that was used to encrypt the value is saved in `AgentHome/conf/agent.scu`.

If you encrypt another property value, the key in `AgentHome/conf/agent.scu` will be used.

Note that after you encrypt `agent.setup.camPword` (or any property that the agent uses to connect to the server) the agent must be able to access `AgentHome/conf/agent.scu` or it will fail to start up. **Do not delete agent.scu.**

If your agent deployment strategy involves distributed a standard `agent.properties` file to all agents, you must also distribute `agent.scu`. For more information, see [Install Hyperic Agents in Volume \(see page 100\)](#).




If agent.scu is missing...

If a Hyperic Agent's `AgentHome/conf/agent.scu` file is missing, subsequent attempts to run the agent start script (`hq-agent.sh` or `hq-agent.bat`) with the `setup` option will fail. To resolve this problem, you must either:

- Reinstall the agent, or
- Perform these steps:
 1. Stop the agent.
 2. Delete its `/data` directory.
 3. Set `agent.setup.camPword` in `AgentHome/conf/agent.properties` to a plain text value.
 4. Start the agent.

Communication Properties Reference

See Agent Properties for definitions of all agent properties.

 agent.setup.camIP

Description

You can use this property to define for the agent the IP address of the Hyperic Server. The Hyperic Agent reads this value only in the event that it cannot find connection configuration in its `data` directory. Specifying this and other `agent.setup.*` properties is a way to reduce the user interaction required to configure an agent to communicate with the server.

The value can be provided as an IP address or a fully qualified domain name. To identify an server on the same host as the server, set the value to `127.0.0.1`.

If there is a firewall between the agent and server, specify the address of the firewall, and configure the firewall to forward traffic on port 7080, or 7443 if you use the SSL port, to the Hyperic Server.

Default

Commented out, localhost.


 agent.setup.camPort

Description

You can use this property to define for a Hyperic Agent, at first startup after installation, what server port to use for non-secure communications with the server. The agent reads this value only in the event that it cannot find connection configuration in its `data` directory. Specifying this and other `agent.setup.*` properties is a way to reduce the user interaction required to configure an agent to communicate with the server.

Default

Commented out, 7080.

 agent.setup.camSSLPort

Description

You can use this property to define for the Hyperic Agent, at first startup after installation, what server port to use for SSL communications with the Hyperic Server. The agent reads this value only in the event that it cannot find connection configuration in its `data` directory. Specifying this and other `agent.setup.*` properties is a way to reduce the user interaction required to configure an agent to communicate with the server.

Default

Commented out, 7443.

 agent.setup.camSecure

Description


You can use this property to define for the agent, at first startup after installation, whether to communicate with the server over SSL. If you set this property to yes, all agent-server communications will be use the SSL secure port.

If acceptable in your environment, non-SSL communication offers improved performance for agent-server communications.

The agent reads this value only in the event that it cannot find connection configuration in its data directory. Specifying this and other agent.setup.* properties is a way to reduce the user interaction required to configure an agent to communicate with the server.

Default

Commented out, value of yes.

 agent.setup.camLogin

Description

You can use this property to define for the Hyperic Agent, at first startup after installation, the Hyperic username to use when registering itself with the server. The permission required on the server for this initialization is Create, for Platforms.

A login from the agent to the server is only required during the initial configuration of the agent.

The agent reads this value only in the event that it cannot find connection configuration in its data directory. Specifying this and other agent.setup.* properties is a way to reduce the user interaction required to configure an agent to communicate with the server.

Default

Commented out, hqadmin.

 `agent.setup.camPword`

Description

You can use this property to define the password that the Hyperic Agent will use when connecting to the Hyperic Server, so that the agent will not prompt for the user to supply the password interactively at first startup. (This is the password for the user specified by `agent.setup.camLogin`.)

The agent reads this value only in the event that it cannot find connection configuration in its `/data` directory. Specifying this and other `agent.setup.*` properties is a way to reduce the user interaction required to configure an agent to communicate with the server.

Starting in Hyper 4.6.6, the first time you start the Hyperic Agent after installation, if `agent.keystore.password` is uncommented and has a plain text value, the agent will automatically encrypt the property value. If you prefer, you can encrypt these (and other, if desired) property values yourself prior to starting the agent. For more information, see [Encrypt Agent Property Value \(see page 99\)](#).

Default

Commented out, `hqadmin`

 `agent.setup.agentIP`

Description

This specifies the IP address that the Hyperic Server will use to contact the Hyperic Agent. If the agent is on the same host as the server, value of `127.0.0.1` is valid.

If there is a firewall between the server and agent, specify the IP address of the firewall, and configure the firewall to forward traffic intended for the agent to the agent's listen address, which can be configured with `agent.listenIP`.

The agent reads this value only in the event that it cannot find connection configuration in its `data` directory. Specifying this and other `agent.setup.*` properties is a way to reduce the user interaction required to configure an agent to communicate with the server.

Default

As installed, `agent.properties` contains a commented out statement that sets the value to **default**. If you use the `agent.setup.*` properties to supply an agent's configuration at first startup, and uncomment this property and leave the value `default`, the Hyperic Server will contact the agent using the IP address that SIGAR detects on the agent host.

 `agent.setup.agentPort`

Description


This specifies the port (on the IP address configured with `agent.setup.agentIP`) on the Hyperic Agent on which the Hyperic Server will communicate with the agent.

If there is a firewall between the agent and the server, set `agent.setup.agentPort` to the appropriate port on the firewall, and configure the firewall to forward traffic intended for the agent to the agent listen port, which can be configured with.

The agent reads this value only in the event that it cannot find its connection configuration in its `data` directory. Specifying this and other `agent.setup.*` properties is a way to reduce the user interaction required to configure an agent to communicate with the server.

Default

As installed, `agent.properties` contains a commented out statement that sets the value to `*default*`. If you use the `agent.setup.*` properties to supply an agent's configuration at first startup, and uncomment this property and leave the value `*default*`, the Hyperic Server will contact the agent on port 2144, unless SIGAR detects it is not available, in which case another default is selected.

 `agent.setup.resetupToken`


Description

You can use this property to configure a Hyperic Agent to create a new token to use to authenticate with the server at startup. The agent reads this value only in the event that it cannot find connection configuration in its `data` directory. Regenerating a token is useful if the Agent cannot connect to the server because the token has been deleted or corrupted.

Regardless of the value of this property, an agent will generate a token the first time it is started after installation.

Default

As installed, `agent.properties` contains a commented out statement that sets the value to `"no"`.


 `agent.setup.unidirectional`

Available only in vFabric Hyperic

Description

Enables the unidirectional communications between the Hyperic Agent and Hyperic Server in vFabric Hyperic. For more information, see [Configure Unidirectional Agent - Server Communication \(see page 112\)](#).

Note that a for a unidirectional agent with a user-managed keystore, you must configure the keystore name in `agent.properties`. See `agent.keystore.alias`.

 About unidirectional communication

If you configure an agent for unidirectional communication, all communication with the server is initiated by the agent. You can configure unidirectional communication at first agent startup, or with the `agent.setup.unidirectional` property in `agent.properties`. Related topics:

- [About Agent - Server Communication \(see page 15\)](#)
- [Hyperic Security Features and Recommendations \(see page 7\)](#).

Default


Commented out, defaults to no.

 agent.keystore.path

Description

This property configures the location of a Hyperic Agent's (version 4.6 or later) SSL keystore. Specify the full path to the keystore. Define the password for the keystore using the `agent.keystore.password` property.

The values of `agent.keystore.path` and `agent.keystore.password` can only be supplied by defining them in `agent.properties`.

 Specifying keystore path on Windows

On Windows platforms, specify the path to the keystore with Unix-style syntax. To use specify a full Windows path:


- replace back slashes with forward slashes
- put a forward slash at the beginning of the path (before the drive letter)
- if the path contains spaces, put a backslash before each space in the path

For example, to specify this Windows path using Unix syntax:

```
C:\Documents and Settings\Desktop\keystore
```

change it to:

```
/C:/Documents\ and\ Settings/Desktop/keystore
```

 Best Practices for Hyperic Agent Keystore

Please see:

- [Hyperic Security Features and Recommendations \(see page 7\)](#)
 - [Configure SSL Options \(see page 35\)](#)
-

Default

AgentHome/data/keystore

 agent.keystore.password

Description

This property configures the password for a Hyperic Agent's SSL keystore. Define the location of the keystore using the `agent.keystore.path` property.

These values of `agent.keystore.path` and `agent.keystore.password` can only be supplied by defining them in `agent.properties`.

Starting in Hyper 4.6.6, the first time you start the Hyperic Agent after installation, if `agent.keystore.password` is uncommented and has a plain text value, the agent will automatically encrypt the property value. If you prefer, you can encrypt these (and other, if desired) property values yourself prior to starting the agent. For more information, see [Encrypt Agent Property Value \(see page 99\)](#).

Password Requirement for Hyperic Keystores

The Hyperic Server's keystore password and private key password **must** be the same — otherwise, the Hyperic Server's internal Tomcat-based server will be unable to start. For information about why, see "<http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html>". Follow the same convention for a Hyperic Agent keystore — set the password for the agent keystore be the same as the agent private key,

Best Practices for Hyperic Keystores

Please see:

- [Hyperic Security Features and Recommendations \(see page 7\)](#)
- [Configure SSL Options \(see page 35\)](#)

Default

none

 agent.keystore.alias

Description

For agents set up for unidirectional communication with the Hyperic Server, the `agent.keystore.alias` property configures the name of the user-managed keystore for the agent. By default, the agent looks for keystore named "hq". For unidirectional agents with user-managed keystores, you must define the keystore name with `agent.keystore.alias`.

For example, given this user-managed keystore for a unidirectional agent:

```
(hq self-signed cert), Jul 27, 2011, trustedCertEntry,
Certificate fingerprint (MD5): 98:FF:B8:3D:25:74:23:68:6A:CB:0B:9C:20:88:74:CE
hq-agent, Jul 27, 2011, PrivateKeyEntry,
Certificate fingerprint (MD5): 03:09:C4:BC:20:9E:9A:32:DC:B2:E8:29:C0:3C:FE:38
```

Define the name of the keystore like this:

```
agent.keystore.alias=hq-agent
```

If the value of `agent.keystore.alias` does not match the keystore name, agent-server communication will fail.


 About unidirectional communication

If you configure an agent for unidirectional communication, all communication with the server is initiated by the agent. You can configure unidirectional communication at first agent startup, or with the `agent.setup.unidirectional` property in `agent.properties`. Related topics:

- [About Agent - Server Communication \(see page 15\)](#)
- [Hyperic Security Features and Recommendations \(see page 7\)](#)

Default

hq

 agent.proxyHost


Description

The host name or IP address of the proxy server that the Hyperic Agent must connect to first when establishing a connection to the Hyperic Server. Supported in vFabric Hyperic only, for agents configured for unidirectional communication.

Use in conjunction with `agent.proxyPort` and `agent.setup.unidirectional`.

Default

none

 agent.proxyPort

Description

The port number of the proxy server that the Hyperic Agent must connect to first when establishing a connection to the Hyperic Server. Supported in vFabric Hyperic only, for agents configured for unidirectional communication.

Use in conjunction with `agent.proxyHost` and `agent.setup.unidirectional`.

Default

none

Encrypt Agent Property Value

Starting in Hyperic 4.6.6, the `agent.properties` file supports encrypted property values.

If, prior to first agent startup, you uncomment and assign a plain text value to `agent.setup.camPword` or `agent.keystore.password`, the agent will automatically encrypt the property value, as described in [Hyperic Security Features and Recommendations](#) (see page 7).

If you prefer, you can encrypt these (and other, if desired) property values yourself.

About Where the Agent Finds Server Connection Data

Note that upon first successful connection to the Hyperic Server, a Hyperic Agent saves the credentials it used in its `/data` directory. Upon each restart, the agent looks first in that directory for server connection details. Hence, edits to the username and password (`agent.setup.cam.Login` and `agent.setup.camPword`) configured in `agent.properties` have no effect, if the agent has valid connection data its `/data` directory.

To add an encrypted entry to `agent.properties`, run the agent start script (`AgentHome/bin/hq-agent.sh` or `AgentHome/bin/hq-agent.bat` with the new `set-property` option, and supply the name of the property and the value you wish to encrypt.

Do not use set-property option on an agent upgraded to v4.6.6

The `set-property` option is only supported for newly installed agents. You cannot manually encrypt properties for an agent that you upgraded to 4.6.6 by pushing the 4.6.6 bundle from the Hyperic Server. Note however that if an upgraded agent's `agent.properties` file contains uncommented password properties with plaintext values, they will be automatically encrypted.

The command syntax is:

```
./hq-agent.sh set-property PropertyKey PropertyValue
```

For example, to set the `agent.setup.camPword` to "hqadmin":

```
./bin/ hq-agent.sh set-property agent.setup.camPword hqadmin
```

If the properties file does not already define the property, the property definition is added at the end of the `agent.properties` file; the encrypted value (not plain text) is shown. For example:

```
agent.setup.camPword=ENC(gaSh3I8gg1olL1EDHHJo/g==)
```

The key that was used to encrypt the value is saved in `AgentHome/conf/agent.scu`.

If you encrypt another property value, the key in `AgentHome/conf/agent.scu` will be used.

Note that after you encrypt `agent.setup.camPword` (or any property that the agent uses to connect to the server) the agent must be able to access `AgentHome/conf/agent.scu` or it will fail to start up. **Do not delete `agent.scu`.**

If your agent deployment strategy involves distributed a standard `agent.properties` file to all agents, you must also distribute `agent.scu`. For more information, see [Install Hyperic Agents in Volume](#) (see page 100).

 If `agent.scu` is missing...

If a Hyperic Agent's `AgentHome/conf/agent.scu` file is missing, subsequent attempts to run the agent start script (`hq-agent.sh` or `hq-agent.bat`) with the `setup` option will fail. To resolve this problem, you must either:

- Reinstall the agent, or
- Perform these steps:
 1. Stop the agent.
 2. Delete its `/data` directory.
 3. Set `agent.setup.camPword` in `AgentHome/conf/agent.properties` to a plain text value.
 4. Start the agent.

Install Hyperic Agents in Volume

- [Establish Installation Environment](#) (see page 100)
 - [Set up an Install Server](#) (see page 100)
 - [Establish Environment on Target Platforms](#) (see page 100)
- [Create Standard Agent Properties File](#) (see page 101)
- [Perform Remote Agent Installations](#) (see page 101)
 - [Install and Start Agents One-by-One](#) (see page 101)
 - [Deploy and Start Multiple Agents at Once](#) (see page 101)
- [Verify Successful Agent Startup](#) (see page 102)

About this page...

This page has recommendations for how to deploy agents in volume in a large Hyperic environment. Before performing the steps below, see [Hyperic Installation and Startup Process](#) (see page 20) for information about prerequisite steps. The instructions below are one way to perform [Step 4 - Set Up the Hyperic Agent](#) (see page 24) of that process.

Establish Installation Environment

Set up an Install Server

Choose a machine that can access all target platforms from which to perform remote installation. We refer to this as the "install server". On the install server, create a user account, for instance "hyperic", with permissions required to SSH into each target platform without a password.

Establish Environment on Target Platforms

On each platform on which you will install the Hyperic Agent:

- Create an user account that is identical to the one you created on the install server.
- Create identical installation directories, for example, `/home/hyperic`.
- Set up a trusted keystore, if desired. See [SSL Setup for New Hyperic Installations \(see page 35\)](#) on [Configure SSL Options \(see page 35\)](#) for recommendations.

Create Standard Agent Properties File

To enable mass agent deployment, you create an `agent.properties` that defines the agent properties required for the agent to start up and connect with the Hyperic Server. If you supply the necessary information in the properties file, each Hyperic Agent will find its setup configuration at startup, rather querying for it interactively.

For example, you can create a standard agent profile that you can copy to the agent installation, or to a location available to the agent installation.

In a standard `agent.properties` — one that you can deploy to multiple agents that report to the same Hyperic Server, you do not edit the properties that specify an agent's listen address and port. At first startup, if explicit values for IP address and port are not set, the Hyperic Agent - which detects the network interfaces on the platform - uses the first detected interface as its listen address, and port 2144 or 2443 as its listen port, depending on whether you configure the agent for plain text or SSL communications.

At a minimum, you must define the Hyperic Server address and port. In addition, you can configure optional agent behaviors that are controlled by agent properties. For more information, see [Set Up Agent in Properties File \(see page 84\)](#).

The first time you start the agent, it will read its properties file for the server connection information, connect to the server, and register itself.

Encrypted values in agent.properties file

If your standard `agent.properties` contains any encrypted values, you must distribute the file that contains the encryption key, in addition to `agent.properties`. The encryption key is stored in the `agent.scu` file in the `AgentHome/conf/` directory of the agent installation where the encryption was performed. The `agent.scu` file must be installed in the same directory on each agent you deploy with the standard properties file.

Perform Remote Agent Installations

Install and Start Agents One-by-One

Follow these steps to install agents one-by-one. To install to all target platforms at once, see [Deploy and Start Multiple Agents at Once \(see page 101\)](#) below.

1. Log on to your installation account on the install server.
2. SSH to the remote platform.
3. Copy the agent archive to the agent host.
4. Unpack the agent archive.
5. Copy the `agent.properties` file to the `/.hq` directory under the home directory of the standard agent installation user account.
6. Start the new agent.

Deploy and Start Multiple Agents at Once

Follow these step to install and start the agent on multiple hosts:

1. Create a `hosts.txt` file on your install server that maps hostname to IP address for each platform to which you wish to install the agent.
2. Open a command shell on the install server.
3. Entering the following commandd in the shell, supplying the correct name of the agent package in the export command:

```
$ export AGENT=hyperic-hq-agent-4.6.0-x86-linux.tgz
$ for host in `cat hosts.txt`; do scp $AGENT $host:</path/to/agent/install> && ssh $host "tar
zxf $AGENT && ./hyperic-hq-agent-4.6.0/hq-agent.sh start"; done
```

If target hosts have sequential names (for example, `host001`, `host002`, `host003`, etc), you can skip the `hosts.txt` file and use the `seq` command like this:

```
$ export AGENT=hyperic-hq-agent-4.6.0-x86-linux.tgz
$ for i in `seq 1 9`; do scp $AGENT host$i: && ssh host$i "tar zxf $AGENT &&
./hyperic-hq-agent-4.6.0/hq-agent.sh start"; done
```

Verify Successful Agent Startup

After registering itself with the Hyperic Server, an Hyperic Agent runs an auto-discovery scan, and should discover its host platform, and supported managed products running on the platform. Check the **Auto-Discovery** portlet in the Hyperic Dashboard to verify the the platforms were discovered.

If you have problems, see [Troubleshoot Agent and Server Problems \(see page 116\)](#).

Upgrade Hyperic Agent

- [Agent Upgrade Methods](#) (see page 103)
- [Push Agent Bundle from the vFabric Hyperic Server](#) (see page 103)
- [Upgrade Agent Bundle Manually](#) (see page 103)
- [Create a Custom Agent Upgrade Bundle](#) (see page 104)
- [Upgrade Agent Using a Full Agent Package](#) (see page 105)

About this page...

This page has instructions for upgrading the Hyperic Agent to 4.6. Before performing the steps on this page, see [Hyperic Upgrade Processes](#) (see page 28) for information about upgrade options and other steps in the upgrade process.

If you want to configure your 4.6 agents for a user-managed keystore, see the information in the [Upgrade Server and Agent with User-Managed Keystores](#) (see page 29) section of [Hyperic Upgrade Processes](#) (see page 28).

Agent Upgrade Methods

To upgrade a Hyperic 4.x Agent to 4.6 you can either:

- Install just the 4.6 agent bundle, by pushing it from your upgraded Hyperic 4.6 Server to each agent (in vFabric Hyperic only) as described in [Push Agent Bundle from the vFabric Hyperic Server](#) (see page 103), or by manually copying the bundle to each agent as described in [Upgrade Agent Bundle Manually](#) (see page 103). If desired, you can first create your own agent bundle, as described in [Create a Custom Agent Upgrade Bundle](#) (see page 104).
- Install complete 4.6 agent package, as described in [Upgrade Agent Using a Full Agent Package](#) (see page 105).

Push Agent Bundle from the vFabric Hyperic Server

Available only in **vFabric Hyperic**

You can update one or more Hyperic Agents by pushing the new bundle to it from the Hyperic Server, using the Hyperic user interface. The bundle must reside in the `ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles` directory.

The agent upgrade command is available on the **Views** tab for an Hyperic Agent (or a group of agents). For more information see [Start, Stop, and Other Agent Operations in vFabric Hyperic Configuration](#).

Note: When you update an agent bundle, the configuration settings in the agent's `AgentHome/conf/agent.properties` file are not changed. However, the first time you start an agent that you have updated to 4.6.6 or later, passwords specified in the file will be encrypted.

Agent Name Unchanged by Bundle Upgrade Process

Note that if you upgrade a 4.x agent by pushing a new bundle from the Hyperic Server, the name of the agent is not changed. So, an agent name that contains the Hyperic version number — as is the default naming convention — reflects the originally installed version, rather than the version to which it has been upgraded. For example, if you push a 4.6 bundle to an agent whose name is "HQ Agent 4.5", the agent name will remain "HQ Agent 4.5".

Upgrade Agent Bundle Manually

Follow these steps if you wish to manually upgrade the agent bundle in your agent installation, instead of pushing the bundle from the Hyperic Server.

Note: When you update an agent bundle, your previous agent configuration is preserved — the `AgentHome/conf/agent.properties` file is not overwritten.

1. Copy the agent bundle (`agent-4.x.y-nnn.tgz` or `agent-4.x.y-nnn.zip`) from

```
ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles
```

to

```
AgentHome/bundles
```

2. Unpack the agent bundle.
3. Edit the `rollback.properties` file in `AgentHome/conf` to specify the location of the new agent bundle and the bundle it will supersede.

Property	Description	Example
<code>HQ_AGENT_BUNDLE</code>	Name of directory with the new bundle, without full path specification.	<code>agent-4.6.0-EE-<i>nnn</i></code>
<code>HQ_AGENT_ROLLBACK_BUNDLE</code>	Name of directory with the old bundle (the one you are upgrading from), without full path specification.	<code>agent-4.5.0-EE-<i>nnn</i></code>

4. Restart the agent. For instructions, see *Start, Stop, and Other Agent Operations* in *vFabric Hyperic Configuration*.

If the upgrade to the new agent bundle fails, an attempt will be made to start the agent using the old agent bundle.

You can determine whether the upgrade was successful and what version you are running by looking at the log files in `AgentHome/logs`.

Create a Custom Agent Upgrade Bundle

This section describes how to create a custom agent bundle. Pre-configuring the agent eases the process of upgrading multiple agents. For additional information, see [Install Hyperic Agents in Volume \(see page 100\)](#).

1. Back up an existing agent located in:

```
ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles
```

- For example:

```
cp  
ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles/agent-4.5.0-EE-nnn.tgz $
```


2. Extract the bundle. For example:

```
tar xzf
ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles/agent-4.5.0-EE-nnn.tgz
```

- This results in a new directory corresponding to the agent bundle, like this:

```
ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles/agent-4.5.0-EE-nnn
```

3. Update the contents of expanded directory. For instance, you could add custom plugins to the plugins directory:

```
ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles/agent-4.5.0-EE-nnn/pdk/plugins
```

4. Rename expanded directory to the name of custom agent bundle. For example:

```
mv ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles/agent-4.5.0-EE-nnn
ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles/my-bundle
```

5. Pack up agent bundle, using the directory name from from the previous step as the tarball file name. For example:

```
tar cvf ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles/my-bundle.tar
ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles/my-bundle;
gzip ServerHome/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-agent-bundles/my-bundle
```

Upgrade Agent Using a Full Agent Package

These instructions apply to both Hyperic HQ and vFabric Hyperic.

To upgrade an Hyperic 4.x Agent using full the agent package:

1. Stop the 4.x agent.
2. Preserve the existing agent configuration.
 - Back up the `agent.properties` file from your previous installation. The default location for `agent.properties` in 4.x installations is the `AgentHome/conf` directory.
 - Note: In some Hyperic environments, `agent.properties` is stored in an alternative location that eases the process of automating the deployment of multiple agents. On Unix-based platforms, that location is the `.hq` subdirectory of the home directory of the user that runs the Agent. If your agent configuration is stored in that location, it will not be over-written by the new installation.
3. If the agent runs on Windows, uninstall the agent service from a command shell in `AgentHome/bin`:

```
hq-agent.bat remove
```

4. Unpack the 4.y agent into the agent installation directory.
5. Restore the backed up `agent.properties` file to `AgentHome/conf` (unless you keep the properties file in the `.hq` subdirectory of the home directory of the user that runs the agent.)

6. Start the Hyperic Agent

- On Unix-like platforms, enter this command in a shell:

```
sh hq-agent.sh start
```

- On Windows, install the new agent service and start the agent. In a command shell in AgentHome/bin enter:

```
hq-agent.bat install
hq-agent.bat start
```

Upgrade Hyperic Server

Topics marked with * relate to features available only in vFabric Hyperic.

- [What Happens During Server Upgrade](#) (see page 106)
- [Upgrade Hyperic Server on Unix-Based Platforms](#) (see page 106)
- [Upgrade Hyperic Server on Windows Platforms](#) (see page 107)
- [Scaling Hyperic Server in vFabric Hyperic 4.6.5](#) (see page 108)
- [Solving Problems with Upgraded Servers with an Oracle Database](#) (see page 108)

About this page...

This page has instructions for upgrading the Hyperic Server to 4.6. Before performing the steps on this page, see [Hyperic Upgrade Processes](#) (see page 28) for information about upgrade options and other steps in the upgrade process.

If you plan to also upgrade your Hyperic Agents to 4.6, and to configure the server and agents for user-managed keystores, see the information in the [Upgrade Server and Agent with User-Managed Keystores](#) (see page 29) section of [Hyperic Upgrade Processes](#) (see page 28).

What Happens During Server Upgrade

You upgrade the Hyperic Server using the Hyperic installer, using the upgrade option. (The installer does not upgrade the Hyperic Agent.)

The installer installs a new version of Hyperic Server; it obtains the configuration information from your previous server installation configuration files and configures the new server instance accordingly.

New Properties in vFabric Hyperic 4.6.5

In Hyperic 4.6.5, there are two new properties in `server.conf` — `tomcat.maxthreads` and `tomcat.minsparethreads`.

If you use Hyperic's internal database, the installer creates a new database instance that contains the data from the existing instance. The new instance has an updated schema, but the PostgreSQL server itself is not upgraded to a new version.

If you use an external database, the installer updates the existing instance.

Upgrade Hyperic Server on Unix-Based Platforms

1. Stop the current server instance. For example:

```
/opt/hyperic/server-4.5.0/bin/hq-server.sh stop
```

2. **If you use an external Hyperic database, back it up before proceeding.**
3. Archive your old Hyperic Server directory, so that if you want, you can revert to the previous version. For example:

```
tar -zcvf hq-server-4.5.0-archive.tgz hq-server-4.5.0-EE
```

4. Run the Hyperic installer in upgrade mode. For example:

```
/opt/hyperic/hyperic-hq-installer/setup.sh -upgrade
```

5. You are prompted to acknowledge the VMware license agreement.
6. The installer prompts for the path to the previous Hyperic Server instance. Enter the path, for example:

```
/opt/hyperic/server-4.5.0
```

7. The installer prompts for the path to the new server instance. Enter the path to the directory under which the new server instance will be installed. For example, to install the new instance under your existing Hyperic home directory:

```
/opt/hyperic
```

- The installer will finish the upgrade.
8. Start the new server instance. For example:

```
/opt/hyperic/server-4.6.0/bin/hq-server.sh start
```

9. In vFabric Hyperic, you can set a variety of sizing properties by running the installer with the `-updateScale` option. For more information, see [Scaling Hyperic Server in vFabric Hyperic 4.6.5](#) (see page 108).

Upgrade Hyperic Server on Windows Platforms

1. Stop the existing server instance using the Windows Services Control Panel.
2. Follow the instructions that apply, depending on whether you use the Hyperic built-in database or an external database:
 - If you use the built-in Hyperic database, the upgrade process will migrate your database schema to the latest edition. Note that PostgreSQL itself is not upgraded to the latest version that ships with Hyperic. The database server remains the one installed when you first installed Hyperic Server.
 - **If using an external database, back it up.**
3. Archive your previous Hyperic Server directory so that if you wish you can revert to the previous version.

4. Run the Hyperic installer in upgrade mode:

```
c:\hyperic\hyperic-hq-installer\setup.bat -upgrade
```

5. You are prompted to acknowledge the VMware license agreement.
6. The installer prompts for the path to the previous Hyperic Server instance. Enter the full path to your existing server installation, for instance:

```
c:\hyperic\server-4.5.0
```

7. The installer prompts for the path where the upgrade version should be installed. Enter the path to the directory that will contain the new server installation. For instance, to install the new instance under your existing Hyperic home directory:

```
c:\hyperic\
```

- The installer will finish the upgrade.

8. Update the Windows Service with the new version information:

```
c:\hyperic\server-4.5.0\bin\hq-server.bat install
```

9. Start the upgraded Hyperic Server using the Windows Services Control Panel.
10. In vFabric Hyperic, you can set a variety of sizing properties by running the installer with the `-updateScale` option. For more information, see [Scaling Hyperic Server in vFabric Hyperic 4.6.5](#) (see page 108).

Scaling Hyperic Server in vFabric Hyperic 4.6.5

Available only in **vFabric Hyperic**

Starting in vFabric 4.6.5, you can set an installation profile that sets the values of key server properties to tune the Hyperic Server based on the size of your environment. For information about the effect of setting an installation profile, see [About Sizing Profiles in vFabric Hyperic](#) (see page 18). For instructions on how to set the installation profile for Hyperic Server after installation or upgrade, see [Change vFabric Hyperic Server Sizing Profile](#) (see page 111).

If you manage more than 50 platforms, and have previously tuned the server properties whose values are set by changing the sizing profile, rather than update the server's sizing profile, you may wish to manually change the values of the server properties that are new in Hyperic 4.6.5 — in an upgrade installation `tomcat.maxthreads` and `tomcat.minsparethreads` are added to `server.conf` with values appropriate for managing less than fifty platforms.

Solving Problems with Upgraded Servers with an Oracle Database

If you are upgrading an Hyperic installation with an Oracle backend and you experience any of the following errors during upgrade, follow the steps below to resolve the problem.

```
Error updating EAM_SERVICE.SERVICE_TYPE_ID: java.sql.SQLException: ORA-02296:
cannot enable (HQDBUSER.) - null values found
Error executing statement desc=null SQL=[
ALTER TABLE eam_stat_errors DROP CONSTRAINT rt_errs_fk_rstat CASCADE
] java.sql.SQLException: ORA-02443: Cannot drop constraint - nonexistent
constraint
```

Fix this with these steps:

1. Restore your database from backup.
2. Execute this SQL:

```
DELETE FROM EAM_SERVICE WHERE SERVICE_TYPE_ID IS NULL;
```

3. Re-run the upgrade.

Uninstalling an Agent

If the agent is managed by Hyperic, remove the platform for the agent before uninstalling it. Then, simply delete the agent's installation folder.

If the agent is installed the agent as a Windows service, run `hq-agent .bat remove` to remove the Windows service.

Change vFabric Hyperic Server Sizing Profile

Available only in **vFabric Hyperic**

In vFabric Hyperic 4.6.5 and later, you can set the values of Hyperic Server resource-related properties by selecting a small, medium, or large *sizing profile*, based on the number of platforms you will manage. The platform thresholds for each profile and server property values for small, medium, and large installation are defined on [About Sizing Profiles in vFabric Hyperic](#) (see page 18).

When you perform a new server installation by running the Hyperic installer with the `-full` qualifier, the installer prompts you to select a sizing profile, as described in [Run Hyperic Installer](#) (see page 53).

You can also select an installation profile after installing the Hyperic Server, or for a Hyperic Server you have upgraded to 4.6.5, by running the Hyperic installer with the `-updateScale` qualifier.

1. Open a command shell on the Hyperic 4.6.5 Server host.
 - On Unix-based platforms, enter:

```
PathToInstaller/setup.sh -updateScale
```

- On Windows platforms, enter:


```
PathToInstaller\setup.bat -updateScale
```

2. The installer will prompt:

```
Please enter the installation profile (small, medium, large)
```

3. Enter the desired profile, for instance:

```
medium
```

 If you use the embedded PostgreSQL database, rather than an external database (required for production environments), select the "small" sizing profile. The "medium" and "large" profiles are *not* supported unless you use an external Hyperic database.

4. The installer will prompt:

```
Please enter the current server installation directory
```

Enter the full path to the server installation directory.

5. The installer will confirm that the profile has been updated, for example:

```
The server is now updated to "medium" profile...
```

6. Restart the Hyperic Server to make the changes take effect.

Configure Unidirectional Agent - Server Communication

Available only in **vFabric Hyperic**

If your security policies dictate, you can configure the agent to initiate all communications with the HQ Server. You can configure unidirectional communications at first startup. Unidirectional communications are always via SSL.

This section has instructions for changing agent communications from bidirectional to unidirectional and vice versa after the agent has already been configured.

- [Changing from Bidirectional to Unidirectional Communications \(see page 112\)](#)
- [Changing from Unidirectional to Bidirectional Communications \(see page 112\)](#)

Changing from Bidirectional to Unidirectional Communications

1. Stop the agent.
2. Remove the agent's `\data` directory.
 - Removing the `\data` directory will cause the agent, at next startup, to look for the startup settings it needs to connect to the HQ Server in its `agent.properties` file; if the properties file doesn't contain them, it will prompt for settings in the shell.
3. Configure the agent for unidirectional communications using one of these methods:
 - If your practice is to provide all agent startup properties in the properties file, edit `agent.properties` to set `agent.setup.unidirectional=yes`, and start the agent.
 - If your practice is to configure the agent startup properties interactively, start the agent, and respond "yes" when asked if the agent should be configured for unidirectional communications.
4. In the HQ user interface, navigate to the platform's Inventory tab and click **Edit** in the "Type & Network Properties" section.
 - In the edit view for "Type & Network Properties", the "Agent Connection" drop-down list will show your currently selected port for bidirectional communications, something like 10.2.0.213:2144, where 10.2.0.213 is the IP address of the platform, and 2144 is the bidirectional port number previously used.
5. Expand the drop-down list and select the entry that shows the same IP address, and "-1" as the port:
6. 10.2.0.213:-1
 - Your agent will now use unidirectional communications.

Changing from Unidirectional to Bidirectional Communications

1. Stop the agent.
2. Remove the agent's `data` directory.
 - Removing the `data` directory will cause the agent, at next startup, to look for the startup settings it needs to connect to the HQ Server in its `agent.properties` file; if the properties file doesn't contain them, it will prompt for settings in the shell.
3. Configure the agent for bidirectional communications using one of these methods:
 - If your practice is to provide all agent startup properties in the properties file, edit `agent.properties` to set `agent.setup.unidirectional=no`, and start the agent.
 - If your practice to configure the agent startup properties interactively, start the agent, and when prompted for communications direction, respond "no" when asked if the agent should be configured to run in uni-directional mode.
4. In the HQ user interface, navigate to the platform's Inventory tab and click **Edit** in the "Type & Network Properties" selection.

5. Select the appropriate agent in the "Agent Connection" drop down.
 - In the edit view for "Type & Network Properties", the "Agent Connection" drop-down list will show your currently selected port for unidirectional communications, something like 10.2.0.213:-1, where 10.2.0.213 is the IP address of the platform, and -1 is the port number.
6. Expand the drop-down list and select the entry that shows the same IP address, and "2144" as the port (or the port you are configured to use, if not the default), for example, 10.2.0.213:2144
 - Your agent will now use bidirectional communications.

Install or Configure vFabric Hyperic License

- [License Consumption in vFabric Hyperic \(see page 114\)](#)
- [Install or Configure License \(see page 114\)](#)

License Consumption in vFabric Hyperic

vFabric Hyperic is priced on a per managed platform basis where a platform is:

- A physical machine or a VM with a Hyperic Agent running on it.
- A network device or network host managed remotely by a Hyperic Agent. (Network Device in *vFabric Hyperic Resource Configuration and Metrics* describes Hyperic functionality for managing remote devices and hosts.)

Install or Configure License



If you obtain vFabric Hyperic as part of vFabric Suite (previously known as "vFabric Platform") refer first to the licensing information and procedure in *Getting Started with vFabric Suite*. If necessary, complete additional licensing tasks in this document.

vFabric Hyperic evaluation distributions include a time-limited license for 60 platforms.

After you purchase vFabric Hyperic, a production license specifies the the number of platforms you may manage, and unless you have a perpetual license, the license expiration date.

To activate your license:

- **If you acquired vFabric Hyperic standalone**, create a file named `vf.hyp-serial-numbers.txt` that contains the product serial number provided by VMware. Install the serial number file in:
 - `/etc/opt/vmware/vfabric/` on Unix-like platforms
 - `%ALLUSERSPROFILE%\vmware\vfabric` on Windows platforms.
- **If you obtained vFabric Hyperic as a part of vFabric Suite**, configure the location of the VMware License Server that administers the network license for vFabric Suite, by adding the `vfabric.licenseServer.url` property to `ServerHome/conf/hq-server.conf`.

After performing the appropriate step above, restart the Hyperic Server.

You can view your license terms on the **HQ License Details** section of the **Administration** tab, as shown in the screenshot below.

vFabric Hyperic sends an email notification of upcoming expiration starting 45 days prior to the expiration date.

Administration

Authentication/Authorization

Users: [List Users](#)
[New User...](#)

Roles: [List Roles](#)
[New Role...](#)

HQ Server Settings

Settings: [HQ Server Settings](#) - Change settings for email, announcements, data compression, database maintenance and authentication services
[Monitoring Defaults](#) - Define monitoring and alerting policies for the entire environment
[Escalation Schemes Configuration](#) - Define global escalation schemes that can be applied to individual alerts
[Plugin Manager](#) - Manage Hyperic product plugins running in your environment.

Plugins

[Network and Host Dependency Manager](#)
[Groovy Console](#)
[HQ Health](#)
[HQ Web Services Api](#)
[tc Server Command-line Interface](#)

HQ License Information

Expires: Mon Oct 10 13:22:44 PDT 2011
Platform Limit: 60
Current Platform Count: 1



Troubleshoot Agent and Server Problems

7

This page has tips for troubleshooting problems in a VMware vFabric™ Hyperic® deployment.

- [Looking for Clues \(see page 117\)](#)
 - [HQ Health \(see page 117\)](#)
 - [hqstats and agentstats \(see page 117\)](#)
 - [Agent Metrics \(see page 117\)](#)
 - [Log Files \(see page 117\)](#)
 - [Thread Dumps \(see page 118\)](#)
 - [Check Port Availability \(see page 119\)](#)
- [Hyperic Server Problems \(see page 119\)](#)
- [Large Event Table \(see page 119\)](#)
 - [License Issues \(see page 119\)](#)
 - [Backlogged Hyperic Server \(see page 119\)](#)
- [Agent Startup or Connection Problems \(see page 120\)](#)
 - [Agent Failed to Connect to Server at First Startup \(see page 120\)](#)
 - [Server Does Not Have Agent Token \(see page 121\)](#)
 - [Agent Start Script Timeout \(see page 121\)](#)
 - [Java Service Wrapper Timeout \(see page 121\)](#)
 - [Problem Running Start Script setup Option \(see page 122\)](#)
- [Invalid or Unknown Availability \(see page 122\)](#)
 - [Out-of-Sync Agent and Server Clocks \(see page 122\)](#)
 - [Overloaded Backend \(see page 123\)](#)
 - [Overloaded Agent \(see page 123\)](#)
- [Slow User Interface \(see page 124\)](#)
- [Warning Messages in the Agent Log \(see page 124\)](#)
 - [Connection Timeout Messages \(see page 124\)](#)

Looking for Clues

This section describes options for getting information that might help you diagnose problems in an Hyperic deployment.

HQ Health

The **HQ Health** page, available in the "Plugins" section of the **Administration** page, displays a variety of metrics and status about your Hyperic deployment, including server host statistics and Hyperic Server process information.

HQ Health provides views, queries, and diagnostic tools that provide visibility into metric loads, caches, the Hyperic database, and agents across your deployment. For more information, see *ui-HQHealth* in *vFabric Hyperic User Interface*.

hqstats and agentstats

Mainly of use to Hyperic Support (or others with an internals-level knowledge of Hyperic), the files in the `hq-server/logs/hqstat` and `AgentHome/logs/agentstats` folders contain a variety of system and subsystem performance, resource usage, and other statistics for the Hyperic Server and Hyperic Agent, respectively.

The statistics files in the `hqstat` and `agentstats` directories are in `.csv` format and can be viewed in a spreadsheet program or other `.csv` viewer.

Agent Metrics

Hyperic Agent metrics are helpful in diagnosing many problems that can occur. By default, these metrics are reported:

- Availability
- JVM Free Memory - Indicator
- JVM Total Memory - Indicator
- Number of Metrics Collected Per Minute - Indicator
- Number of Metrics Sent to the Server Per Minute
- Server Offset
- Total Time Spend Fetching Metrics per Minute

Depending on your environment, you may find it useful to track other agent metrics, such as:

In addition to the default metrics

- Number of Metrics which Failed to be Collected
- Number of Metrics which Failed to be Collected per Minute
- Maximum Time Spent Processing a Request
- Number of Connection Failures
- Total Time Spent Fetching Metrics per Minute

For more information about default and available agent metrics see [View Hyperic Agent Metrics](#).

Log Files

The following log files can be a useful source of information in the event that a problem occurs in a Hyperic deployment:

- `ServerHome/logs/wrapper.log`
- `ServerHome/logs/bootstrap.log`
- `ServerHome/logs/server.log`

- `ServerHome/logs/hqdb.log` (only available for deployments using the built-in PostgreSQL database).
- `AgentHome/logs/wrapper.log`
- `AgentHome/logs/agent.log`
- `AgentHome/logs/agent.log.startup`

You can increase the level of logging an agent performs in its `agent.properties` file. Note that debug logging is very verbose and uses more system resources. Hyperic recommends configuring debug logging only when troubleshooting problems, and only at the subsystem level. For more information, see [Configure Agent Logging](#).

Thread Dumps

This section has instructions for generating thread dumps for the Hyperic Server and Hyperic Agent.

Generate a Hyperic Server Thread Dump from User Interface

Follow these steps to output a server thread dump to your browser.

1. Click the **Administration** tab.
2. Click **HQ Health** in the **Plugins** section of the **Administration** page.
3. Click **Print** in the **HQ Process Information** section on the **HQ Health** page.

Generate a Hyperic Server Thread Dump from Command Line

To generate a thread dump on Windows:

- If Hyperic Server is running in a terminal window — Try `<ctrl><break>` in the terminal window.
- If Hyperic Server is running as a service — Use a tool like "[StackTrace \(http://www.adaptj.com/main/stacktrace\)](http://www.adaptj.com/main/stacktrace)."

To generate a thread dump on Unix-like systems, use `jstack` "


<http://download.oracle.com/javase/1.5.0/docs/tooldocs/share/jstack.html>" on the Hyperic Server process. For example, if the server's PID is 215:

```
jstack 215 >mydumpfile.txt
```

How to find the Hyperic Server PID

You can run `jps` in a shell to determine the Hyperic Server's process ID — look for the process named "Bootstrap". For example:

```
$ jps
187 WrapperStartStopApp
408 Jps
215 Bootstrap
```

-
-  On Unix-like systems, you can also run `kill -3` on the Hyperic server process. Note however, that if you do, the thread dump will be written to `wrapper.log`, and be difficult to parse.
-

Generate Agent Thread Dump from User Interface

Run the agent launcher with the `dump` option. For more information see [Start, Stop, and Other Agent Operations](#)

Check Port Availability

The Hyperic Server must be able to establish a connection with the agent, and vice versa.

To verify that the server can access the agent's listen port, run the following from the server platform:

```
telnet AgentIp AgentPort
```

For example:

```
$ telnet 192.168.1.114 2144
```

For a successful connection, the results are similar to:

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^\\'.
GET
Connection closed by foreign host.
```

To verify that the agent can access the server's listen port, run the following from the agent platform:

```
$ telnet ServerIP ServerPort
```

For example:

```
$ telnet 192.168.1.114 7080
```

Hyperic Server Problems

This section describes problems that could prevent the Hyperic Server from starting up.

Large Event Table

A message like the following in `server.log` indicates that the `EAM_EVENT_LOG` is so large that trigger initialization is taking more than 15 minutes.

License Issues

In vFabric Hyperic, the number of platforms you can manage is limited by your license. For more information, see [Install or Configure vFabric Hyperic License \(see page 114\)](#).

Backlogged Hyperic Server

When the Hyperic Server starts after a period of downtime, it can be inundated with metric reports from agents that continued to run while the server was down. When the server is processing a large metric backlog from many agents, the maximum size of the agent-server connection pool size can become a bottleneck and affect server performance.

You can check the "Current Thread Busy" metric for the Hyperic Server's internal Tomcat server to determine whether connection pool size is an issue.

To enable a restarted server catch up with a metric backlog you can increase the maximum size of the agent-server connection pool — typically this is the only time that increasing the size of the connection pool is indicated. The maximum number of agent-server connections is configured with the `org.hyperic.lather.maxConns` property in `web.xml`. Note however that, in effect, the number of connections is limited by the maximum number of server execution threads, which is configured using the `tomcat.maxthreads` property in `server.conf`. So, when you increase the value of `org.hyperic.lather.maxConns`, it may be necessary to increase the value of `tomcat.maxthreads`.

To enable a backlogged Hyperic Server to catch up, enable 5% to 10% more connections than there are agents reporting to the server. For example, if you have 1000 agents, enable 1050 to 1100 agent-server connections.

To change the size of the agent-server connection pool:

1. As necessary, update the maximum number of agent-server connections — `org.hyperic.lather.maxConns` — in `<Server installation directory>/hq-engine/hq-server/webapps/ROOT/WEB-INF/web.xml` file, in the stanza shown below. Ensure that the value of `maxConns` is 5% to 10% greater than the number of agents that report to the server.

```
<init-param>
<param-name>org.hyperic.lather.maxConns</param-name>
<param-value>3000</param-value>
</init-param>
```

2. As necessary, configure the maximum number of Tomcat threads — `tomcat.maxthreads` — in `server.conf`. Ensure that the value of `maxThreads` greater than the value of `maxConns`.
3. Restart the Hyperic Server to enable the changes to take effect.

Agent Startup or Connection Problems

This section describes problems that could prevent the Hyperic Agent from starting up or connecting to the Hyperic Server.

Agent Failed to Connect to Server at First Startup

Every time an agent starts up it attempts to contact the Hyperic Server. If, the first time you start up an agent, it cannot connect to the server, the agent will continue to have problems connecting to the server, even after the server is reachable.

The first time a Hyperic Agent successfully connects with the Hyperic Server, the agent saves the server connection settings in its `/data` directory. If the server is not available (because the wrong address/port was configured for the agent, or because the server hasn't been started or is still in the process of starting up) the agent will fail to connect, and hence fail to persist the server connection data. Upon agent restart, the agent will not be able to find the connection data it requires, and fail to connect to the server. In this case, `server.log` will contain a message similar to:

```
2010-04-20 11:04:26,640 ERROR [Thread-1|Thread-1] [AutoinventoryCommandsServer] Unable to send
autoinventory platform data to server, sleeping for 33 secs before retrying. Error: Unable to
communicate with server -- provider not yet setup
```

To solve this problem:

1. Delete the agent's `/data` directory.
 - This forces the agent to obtain new agent - server communication properties.

2. Verify the address and listen port for the Hyperic Server.
 - Is there a firewall between agent and server? See the instructions in [Configure Agent - Server Communication Interactively](#) or [Configure Agent - Server Communication in Properties File](#).
3. Verify the Hyperic Server is up.
4. Start the agent, supplying the correct server connection properties, either in `agent.properties` or interactively. See the instructions referenced in step 2 above.

Server Does Not Have Agent Token

Starting in 4.6.5, if all platforms managed by an agent are removed from Hyperic inventory, the Hyperic Server also removes the saved authentication token for that agent from the Hyperic database. So after you delete a platform that is managed by an agent that does not manage any other platforms (as in the case of an agent that manages only the platform it runs on), the Hyperic Server will not accept connections from that agent.

If you want the agent to rediscover the platform, you must repeat the initial agent setup process. To force the agent setup dialog:

- While the agent is running, by running the agent launcher with the `setup` option, or
- By deleting the agent's `/data` and restarting the agent.

Agent Start Script Timeout

By default, the agent start script times out after five minutes if the startup sequence is not successful. Check the `agent.log.startup` file for messages.

If desired, you can configure a longer timeout period – to give the agent more time to connect to the server — by adding the `agent.startupTimeOut` property, defined below, to the `agent.properties` file.



`agent.startupTimeOut`

Description

The number of seconds that the agent startup script will wait before determining that the agent did not startup successfully. If the agent is not determined to be listening for requests within this period of time, an error is logged, and the startup script times out.

Default

As installed, `agent.properties` does not contain a line that defines the value of this property. The default behavior of the agent is to timeout after 300 seconds.

After editing the `agent.properties` file, save your changes and restart the agent.

Java Service Wrapper Timeout

Under high load, the agent may become unresponsive. If this occurs and there are no coincident errors or warnings in the agent log that indicate another explanation, it may be that the agent JVM was starved for memory, and unresponsive to a ping from the Java Service Wrapper (JSW).

In that case the `wrapper.log` file will contain an entry like this:

```
ERROR | wrapper | 2009/01/15 02:15:18 | JVM appears hung: Timed out waiting for signal from JVM.
```

To resolve the problem, you can configure the JSW to give the agent more time to respond to startup and ping requests.

Increase the JSW's timeout period from 30 seconds to 300. To do so, add this property to `AGENT_HOME/bundles/agent-4.x.xxxx/conf/wrapper.conf`.

```
wrapper.ping.timeout=300
```

This will cause the JSW to wait longer for a ping response before determining that the JVM is hung.

Increase the agent's startup timeout from 30 seconds to 300. This will give the agent more time to start up before wrapper gives up on it and kills the process. To do so, add this property value:

```
wrapper.startup.timeout=300
```

to

```
AgentHome/bundles/agent-4.x.xxxx/conf/wrapper.conf
```

Problem Running Start Script setup Option

If agent.scu is missing...

If a Hyperic Agent's `AgentHome/conf/agent.scu` file is missing, subsequent attempts to run the agent start script (`hq-agent.sh` or `hq-agent.bat`) with the `setup` option will fail. To resolve this problem, you must either:

- Reinstall the agent, or
- Perform these steps:
 1. Stop the agent.
 2. Delete its `/data` directory.
 3. Set `agent.setup.camPword` in `AgentHome/conf/agent.properties` to a plain text value.
 4. Start the agent.

Invalid or Unknown Availability

This section describes reasons that Hyperic might incorrectly show an agent (or agent-managed resource) as unavailable, show its availability status as "Unknown" (availability icon is grey), or "flapping" availability values.

Out-of-Sync Agent and Server Clocks

If Hyperic erroneously indicates that resources are unavailable, it may be because the system clocks on the agent and server hosts are out-of-sync. By default, Hyperic monitors the offset between the server and an agent — see the "Server Offset" metric on the **Monitor** tab. An offset of less than one minute is unlikely to pose problems; with a larger offset, problems may occur.

To solve an offset problem, install NTP and synchronize system clocks on the agent and server hosts.

To prevent agent and server becoming significantly out-of-sync, you can run NTP on each system, use Hyperic to monitor the offset on each system, and set alerts based on the offset metric. To do so, configure a platform service of type "NTP" to monitor each NTP service, and set alerts to fire when an offset from the time authority grows unacceptably high. For more information, see the "Network Services" section in *vFabric Hyperic Resource Configuration and Metrics*.

Overloaded Backend

If the Hyperic database cannot process metrics at the rate it receives them, resources that are available may be incorrectly shown as unavailable.

You can view Hyperic Server process statistics, as well as load and utilization data - load, process, system and JVM memory, and so on - on the **HQ Health** page.

Based on your load, it may be appropriate to tune Hyperic Server, the Hyperic database, or both. For more information, see *Scaling and Tuning Hyperic Performance*.

Important: First and foremost: run the Hyperic database on dedicated hardware. Competition for system resources can slow down metric processing.

Overloaded Agent

If an agent's queue of metrics grows to a certain level over a period of time, the following warning message is written to the agent .log file:

```
The Agent is having a hard time keeping up with the frequency of metrics taken. Consider increasing your collection interval.
```

To investigate, you can configure the the agent to report the "Total Time Spent Fetching Metrics per Minute" metric — if the agent spends more than half its time fetching metrics — it is overloaded.

You can alleviate the problem by

- Increase metric collection intervals — For most metrics, the default is every 5 or 10 minutes. You can change the collection interval for all metrics collected for a resource type on the **Administration > Monitoring Defaults** page for the resource type.



Want to Change a Whole Bunch of Metric Collection Intervals?

If you want to change metric templates in bulk, or without using the Hyperic user interface, you can change metric collection settings - including collection intervals - for a resource type with the HQApi **metricTemplate** command. You can use the **metricTemplate sync** option from the command line or in a script, as desired. For more information, see HQApi **metricTemplate** command section in *vFabric Hyperic Web Services API*.

 Try to redistribute load

If an agent that logs metric volume warnings is monitoring a large number of remote services over the network (for example, HTTP, FTP, SNMP, or another service type whose protocol the agent supports), you can spread the load around — configure a different agent to monitor some of the network services. You can compare the agent loads on the **Agents** tab of **HQ Health**.

Slow User Interface

If Hyperic's web user interface is slow, the cause may be an overloaded backend - the Hyperic database, or the Hyperic Server itself. See [Overloaded Backend \(see page 123\)](#).

Warning Messages in the Agent Log

This section has information about the significance of selected warning messages that might be written to the `agent.log` file.

Connection Timeout Messages

Lather, the connection protocol for agent-to-server communication, is configured such that agent connections time out after five minutes, and a timeout message is written to `agent.log`. You can increase the timeout period from 300000 to 900000 in this file:

```
hq-engine/server/default/deploy/lather-jboss.sar/jboss-lather.war/WEB-INF/web.xml
```

in this stanza:

```
<init-param>
<param-name>org.hyperic.lather.execTimeout</param-name>
<param-value>900000</param-value>
</init-param>
```