Traverse On Premise

Reference Guide

Version R92

April 19, 2016

Copyright Agreement

The purchase and use of all Software and Services is subject to the Agreement as defined in Kaseya's "Click-Accept" EULATOS as updated from time to time by Kaseya at http://www.kaseya.com/legal.aspx. If Customer does not agree with the Agreement, please do not install, use or purchase any Software and Services from Kaseya as continued use of the Software or Services indicates Customer's acceptance of the Agreement."

Contents

About This User Guide

This user guide provides information that applies only to Traverse on premise users. Information that applies to both Traverse cloud users and Traverse on premise users is presented in the Traverse User Guide (http://help.kaseya.com/WebHelp/EN/tv/9020000/index.asp#home.htm).

Chapter 1

About Traverse

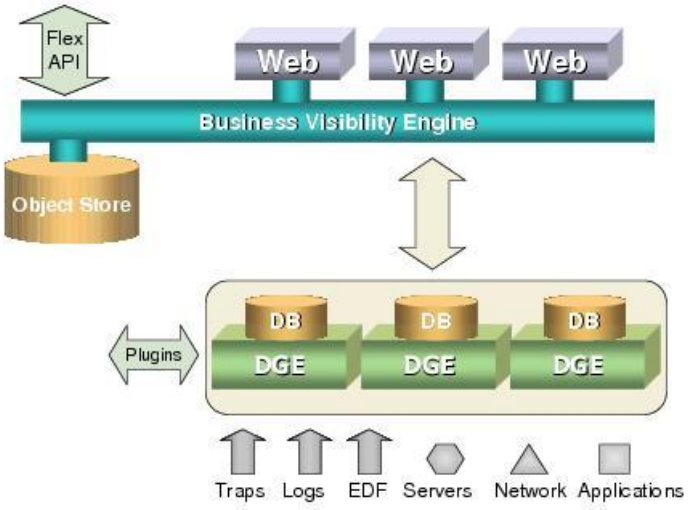
The Traverse architecture allows for a wide range of installation options. This chapter describes the Traverse architecture and its components in detail.

In This Chapter

Traverse Architecture

Traverse uses a three-tier architecture consisting of the Business Visibility Engine (BVE) Web Application, the BVE ObjectStore, and one or more Data Gathering Engines (DGEs). All configuration and authentication information is stored in the BVE ObjectStore (embedded Object Oriented database).

For enterprises, all of the **Traverse** hardware can exist at one central location or be geographically distributed to accommodate multiple offices or divisions. One DGE can monitor all the devices at a single location, including servers, routers, switches, applications, and network appliances. The DGE measures and stores aggregated performance data locally, and forwards only events or alarms to the BVE components. For a data center environment, Kaseya recommends that you employ a DGE at each data center location and set up the provisioning and authentication database at a central location (for example, your NOC).



Traverse System Overview

The **Traverse** system comprises three main components. In a large environment, Kaseya recommends that each component reside on its own host server.

- BVE ObjectStore: An embedded object-oriented database that stores all configuration information. This includes metadata related to user authentication, devices, tests, thresholds for test results, action profiles and other key information. The BVE FlexAPI, which allows access to the BVE for provisioning and results, also operates on this server.
- BVE Web Application: Provides the web-based user interface into **Traverse**. It correlates the data from multiple DGEs, and allows end users to look at the real-time status of their devices, add new devices and actions, and execute reports, using a simple web browser. It manages the distributed databases and distributed processing while generating the real-time reports and graphs. You can have more than one BVE Web Application for load sharing, which allows the use of any load balancing hardware to load-share all access across the Web Applications.
- Data Gathering Engines (DGE): Perform the actual polling of data, receives SNMP traps, generates alarms based on thresholds, and does the aggregation of data in real time. DGEs should be located as close as possible to the devices being monitored to reduce wide area network traffic. The DGEs can be geographically dispersed or you can have multiple DGEs in the same location to distribute the load across different physical servers. When you have multiple DGEs in the same location, the system automatically provisions new devices onto the DGE with a lower number of devices.

Although these architectural components are designed to reside on different servers, **Traverse** allows you to configure two or more components on a single server.

DGEs schedule and perform tests, archive and aggregate data, and trigger notifications and actions. Effective management of historical information is done by setting bounds on storage that prevent it from growing to unmanageable limits. Historical records are aggregated and stored for over a year by default. Historical alarm and event data (changes in severity level) are retained without aggregation and only aged by user selection.

During the initial installation, you can import existing department or device records, or a subset thereof, into the **Traverse** Provisioning Database using the BVE FlexAPI. The system comes with default thresholds for all tests, which you can automatically update using the baselining feature after operating **Traverse** for a few days.

If you are using firewalls within the data center, you must configure access through the firewalls to enable the monitoring of the devices behind them. If the number of devices behind a firewall is significant, you can connect the DGE to a port behind the firewall. Also, if you are using Network Address Translation (NAT) or private address space, the IP address must be unique within the data center.

Overview of System Operation

Each component of **Traverse** operates independently to provide a high level of scalability and fault tolerance. When you start a DGE, it searches the `dge.xml` file for its unique name, which must match the name **you specified when you created the DGE** *(page 30)*. The DGE connects to the BVE ObjectStore—specified in the `emerald.xml` file—and downloads the entire configuration associated with that unique name, including tests, thresholds and actions.

After this process completes, the DGE performs tests, generates events when thresholds are crossed, and triggers the corresponding notifications. The data collected by each DGE is stored in a local SQL database on the DGE itself.

Any configuration changes made in the BVE ObjectStore through the BVE API or the Web Application is instantly transmitted to the appropriate DGE.

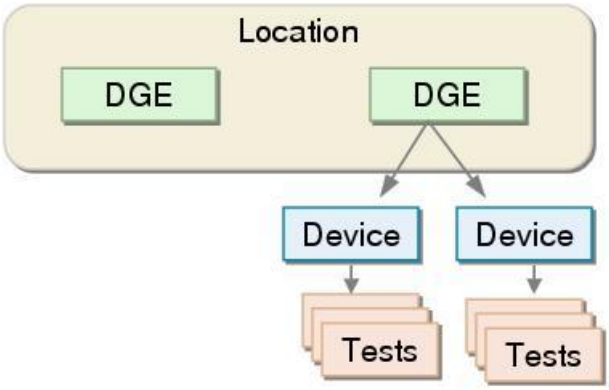
When a user logs into the Web Application, the system searches the Provisioning Database for the list of devices that the user has permission to view. The Web Application then connects directly to the distributed DGEs and gets the real-time status of the services or devices. When the user needs a report, the Web Application fetches the data using parallel queries from the distributed DGEs and generates the reports in real time.

Log files and SNMP traps are managed by the Input Stream Monitor (ISM) and the trap daemon, respectively, and then matched against user-defined regular expressions that define severities and corresponding actions/notifications. You can view these text-based messages in the Message Window in the BVE Web Application.

DGEs and DGE Locations

A DGE is the Data Gathering Engine. The DGE polls devices for various tests such as CPU and bandwidth utilization and aggregates the data that it gathers. Generally, a DGE is physically near the devices that it monitors. A DGE can typically monitor approximately 500 - 1,500 devices. See **Disk Space Requirements for DGE Aggregation** *(page 30)* for sizing algorithms).

A DGE location is a collection of one or more DGEs that are automatically load balanced for provisioned tests. The DGEs within a single DGE Location are usually located in the same physical region, but they can be separate in some special situations. When you provision a device, you assign it to a DGE Location, not to an individual DGE. If there is more than one DGE at that location, **Traverse** automatically assigns the device to the least loaded DGE.



Relationship Between Locations, DGEs, Devices, and Tests

Chapter 2

Installing and Upgrading Traverse On Premise

In This Chapter

Overview

Traverse is a distributed application that comprises three basic software components:

1. BVE Provisioning Database
2. BVE Web Application (User Interface)
3. DGE (Data Gathering Engine)

The BVE Web Application and the BVE Provisioning Database are usually installed on the same server, although you can install the BVE Web Application on a separate server. Depending on the size of your network, you can install all components (including the DGE) on a single server, or you can install the DGE on a separate server. There is only one BVE Provisioning Database for each **Traverse** instance.

As your IT infrastructure expands, you can add new DGEs as required. These DGEs are responsible for monitoring the IT infrastructure and sending alert notifications when a problem is detected. The plugin actions and the plugin monitors allows you to efficiently extend the functionality of the DGEs beyond built-in capabilities.

Note: See **Traverse Quick Start Guide** *(http://help.kaseya.com/webhelp/EN/tv/9020000/EN_TraverseQuickStart_R92.pdf#zoom=70&navpanes=0)* to quickly install and deploy **Traverse**.

System Requirements

Supported Platforms

Windows

- Windows XP Professional with Service Pack 3
- Windows Server 2003 Standard Edition
- Windows Server 2003 Enterprise x64 Edition
- Windows Server 2003 Enterprise x64 R2 Edition
- Windows Server 2008 x64 Edition
- Windows Server 2008 x64 R2 Edition

UNIX

- RedHat Enterprise Linux ES/AS 5 or 6 on x86 platforms
- CentOS 5 or 6 on x86 platforms

Hardware Requirements

For smaller environments (about 100 devices), you can install and operate the entire application from a single server. The minimum hardware reqrements for **Traverse** are:

- 2GHz+ CPU on x86 platform
- 4GB RAM
- 60GB disk space (SCSI or fast IDE)

Recommended configuration:

- 2 x 3GHz+ Intel Xeon CPU (multi-core ok)
- 8GB RAM

- 80GB disk space in RAID-5 configuration (SAS/SATA or SSD)

Some desktop-class processors like the Celeron (which has minimal onboard cache) are not suitable for use with **Traverse**. We strongly recommend Pentium 4/M, Xeon, or equivalent processors.

UNIX Software Requirements

You must install the following software on Linux and Solaris platforms:

- Perl version 5.8 and above programming language/interpreter (available from **http://www.perl.com** *(http://www.perl.com)*).
- Install the Legacy Support Package on computers with the RedHat/CentOS operating system. To install the Legacy Support package, log in to the computer as root and execute following command on the command line

```
yum install "Legacy Software Support"
yum install -y libstdc++.i686 compat-libstdc++-33.i686 popt.i686 zlib.i686
ncurses-devel.i686 glib2.i686
```

Windows Software Requirements

Some anti-virus/malware tools are known to cause database corruption when they attemp to intercept read/write requests. In order to avoid such issues, it is strongly recommended that McAfee, Norton and other anti-virus tools are configured to exclude `<TRAVERSE_HOME>\database` directory from all manual/on-access scans.

Disk Space Requirements

Kaseya recommends 36GB of free space in a RAID 5 configuration be available for the installation of **Traverse**. A minimum of 16GB of free space should be available if the recommended disk requirements cannot be met. See **Disk Space Requirements for DGE Aggregation** *(page 30)* for further requirements.

The BVE Web Application and BVE ObjectStore components have a low impact on disk space. However, these components have a very high impact on CPU performance when processing and generating reports.

Additionally, make sure you plan for the space requirements of the following directories (created during the installation of **Traverse**) when deploying **Traverse**.

Note: References to `<TRAVERSE_HOME>` indicate the top-level directory into which you installed **Traverse**. By default, this is `\Program Files (x86)\Traverse` on Windows and `/usr/local/traverse` on UNIX.

Windows

- `<TRAVERSE_HOME>\database\provisioning` - Provisioning data. Plan for 1MB for every 1000 tests.
- `<TRAVERSE_HOME>\database\mysql` - DGE historical data. See **Disk Space Requirements for DGE Aggregation** *(page 30)* for information about calculating disk space requirements for a DGE database.
- `<TRAVERSE_HOME>\logs` - Plan for 5GB of disk space for log files.

UNIX

- `<TRAVERSE_HOME>/database/provisioning` - Provisioning data. Plan for 1 MB for every 1000 tests.
- `<TRAVERSE_HOME>/database/mysql` - DGE historical data. See **Disk Space Requirements for DGE Aggregation** *(page 30)* for information about calculating disk space requirements for a DGE database.
- `<TRAVERSE_HOME>/logs` - Plan for 5GB of disk space for log files.

Deployment Considerations

Prior to your install, you should ensure that you have complete information about your IT environment where **Traverse** is being installed.

> **Note:** You can specify a port number other than (the default) 80 when installing **Traverse**. Remember to include this port number in the **Traverse** URL.

Traverse Installation Checklist

Question	Relevance
Number of geographical locations with significant concentration of devices?	Instead of geographical locations, you can use the network topology instead. Install a DGE at each location that has a large concentration of devices. Use a single centralized DGE for small remote locations.
Number of devices to be monitored in each location?	This is for sizing the DGE at each location. Each DGE can typically handle 500-1500 devices.
Are there any large switches, routers, or servers at each location?	A large switch with 500 ports can have close to 3000 tests (6 tests for every port). This is the same as the number of tests on 100 devices.
Number of departments accessing the system?	You need to determine the permissions for each department (Read-Only or Read/Write). Also, you need to determine whether departments manage their own devices in Traverse, or whether another centralized department manages these devices.
Are there any existing custom monitors that require migration to Traverse?	Use the various APIs to interface any custom monitoring scripts to Traverse. See the **Traverse Developer Guide & API Reference** *(http://help.kaseya.com/WebHelp/EN/tv/9020000/DEV/index.asp#home.htm)*.
Do you need to interface with any existing provisioning system?	You can add the existing inventory system you use manage devices on the network directly into Traverse.
Are there any other web servers or instances of MySQL operating on the Traverse Server?	Traverse includes its own web server, and you must disable IIS or any other web server operating on the server. Alternatively, configure Traverse to operate on an alternate port. See **Web Server TCP/IP Port** *(page 62)* for more information. Make sure that you disable any firewalls operating on the Traverse server. See Problem: Cannot access Web Application for more information.

Large Environments

For large environments that have at least 30000 to 50000 tests, for 1000 or more devices, Kaseya recommends that you add an additional DGE for monioring for every 800-1200 devices, approximately one DGE for every 20000 tests.

The actual monitoring capacity depends on the number of tests on each device. A server might only have four or five tests, but a large switch with 500 ports can have as many as 5000 tests. If a DGE can no longer manage tests due to high volume, the internal queues begin backing up and a message is automatically sent to the error log.

However, avoid deploying too many DGEs, because it increases administrative overhead and the probability of failures.

An example hardware configuration for a DGE-only server in a large environment is as follows:

- Dual Pentium 4 Xeon (2GHz+)
- 4GB RAM
- 80GB fast SCSI/SATA drives on RAID-5/RAID-10

Static IP Addresses

Because **Traverse** components (on different servers) communicate with each other over TCP/IP protocols, you must configure the servers on which you are installing **Traverse** with static IP address. During the installation process, you are prompted for the IP address of the host w/BVE ObjectStore. When configuring new DGEs in the **Traverse** Web Application or BVE API server, you must specify the corresponding IP addresses.

Using a static IP address ensures proper operation of the communication subsystem service and prevents issues from occurring in BVE/DGE communications.

Installing Traverse

Before you begin installing **Traverse**, make sure that there are no web servers or databases operating on the server. This creates port conflicts that might prevent **Traverse** from starting.

Traverse is distributed as a single self-extracting executable file (`traverse-x.y.z-windows.exe`) for Windows platforms, and a compressed archive (tar.gz) file called `traverse-x.y.z-OS.tar.gz` for UNIX platforms.

In addition to the installation file, you need a license key to use **Traverse**. This can be either a limited-time trial key, or a permanent key based on the terms of your purchase.

Contents of <TRAVERSE_HOME>

The following table lists the contents of the `<TRAVERSE_HOME>` directory:

Directory	Description
`apps/`	Supporting applications required for Traverse.
`bin/`	Utility software for Traverse components.
`database/`	Runtime database for tests and provisioning.
`etc/`	Configuration files and startup scripts.
`lib/`	Component libraries.
`logs/`	Error and debug log files.
`plugin/`	User custom actions and monitors.
`utils/`	Useful utility tools.
`webapp/`	The Web Application.

Note: References to `<TRAVERSE_HOME>` indicate the top-level home directory into which you installed **Traverse**. By default, this is `\Program Files (x86)\Traverse` on Windows and `/usr/local/traverse` on UNIX.

Installing Traverse on Windows

1. Double-click `traverse-x.y.z-windows.exe`.
2. Follow the instructions in the **Traverse** installation program.
3. When the installation is complete, you must reboot the server before you can use **Traverse**.

Installing Traverse on UNIX Platforms

1. Change to a temporary directory with at least 100 MB of disk space:

```
cd /tmp
```

2. Copy the downloaded **Traverse** archive to the temporary directory:

```
cp /download/dir/traverse-x.y.z-platform.tar.gz
```

3. Extract the software package.

> **Note:** (Solaris only) Use the GNU version of tar instead of the native tar utility in the following command.

```
gunzip -c traverse-x.y.z-platform.tar.gz | tar xpf -
```

4. Change to the directory containing the extracted files:

```
cd traverse-x.ysu root
```

5. If you need to make any changes to the software license key, make the changes before executing the installation script. If the terms of your license change—for example, a change in the expiration date or number of devices—**Kaseya Support** *(https://helpdesk.kaseya.com/home)* provides you with a new license file. Save the new key, overwriting any existing key:

```
traverse-x.y/etc/licenseKey.xml
```

6. As root, execute the installation script:

```
su root
sh ./install.sh
```

Upgrading to Traverse R92

The information in this section describes how to upgrade from **Traverse** 5.5 or later to **Traverse** R92. If you want to upgrade to R92 from versions earlier than 5.5, upgrade to 5.5, and then upgrade to R92.

The upgrade to R92 allows you to preserve all configuration and historical performance data.

Before starting the upgrade process, make sure that:

- **Traverse** 5.x is installed and operating properly on the existing servers.
- You have (local) administrator permissions on the servers on which you installed **Traverse**. You need a login account that is a member of the local Administrators group (either directly, or inherited through other group memberships).
- There is sufficient disk space available on the servers. Kaseya recommends that twice the amount of space currently used by the **Traverse** installation directory is available for the upgrade.

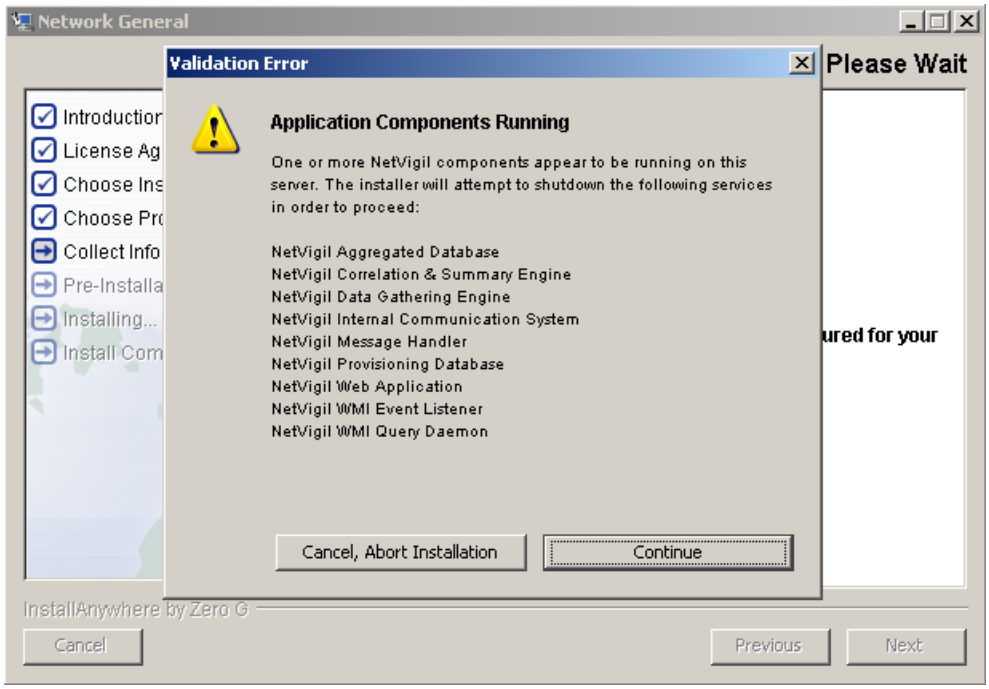
Upgrading Traverse from 5.5 or later to R92 on Windows

1. Log in to **Traverse** as `administrator` or as another user with equivalent permissions.
2. Shut down all **Traverse** components (Start > Programs > Traverse > Stop Traverse Components).

   In a distributed configuration, stop **Traverse** components on the DGE hosts, and then on the server where the BVE/Provisioning Database is operating. In this environment, always make sure you upgrade the system running the BVE/Provisioning Database first.
3. Create a new directory (for example, `C:\OLD-traverse-5.x`).
4. Copy `<TRAVERSE_HOME>` into the directory you just created.
5. Download and save the installation package for **Traverse** R92 (a single executable) to a temporary location on the server.
   - Double-click on the installation executable.
     If **Traverse** 5.x is installed on the system, the installer prompts you to confirm the upgrade to **Traverse** R92.
6. Follow the on-screen prompts to complete the installation.

Components Running Prompt: Windows Upgrade

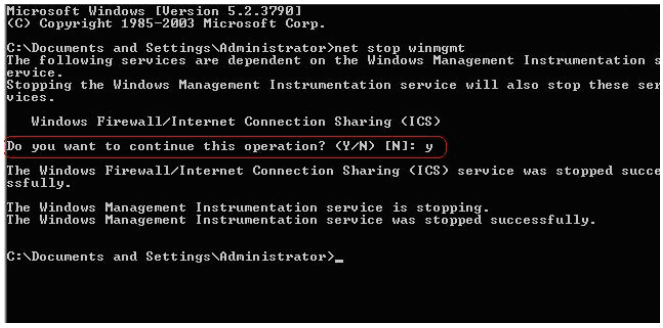
During the installation/upgrade, you might see a message, indicating that the installer cannot access some files in the **Traverse** directory because they are being used by the WMI service.



To resolve this issue and continue the installation, open a command window and enter the following, and then click **Continue**:

```
net stop winmgmt
```

The installer might prompt you to stop other dependent services. If so, stop these services.



After the confirmation prompt displays, continue the installation/upgrade.

> **Note:** Prior to migrating existing configuration and historical data, the installer displays a confirmation prompt.

Review Configuration Files

1. When the installation/upgrade completes, review the configuration files from the previous version of **Traverse** in the `<TRAVERSE_HOME>\OLD-traverse-5.x` directory. If you made any modifications to the configuration files, such as adding a new JDBC driver or configuring plug-in authentication, you must manually re-apply the changes.
   - If you have **Traverse** installed on a single server only, skip to Step 4.
   - If you have **Traverse** installed on more than one server (multiple DGEs, and/or BVE and DGE on separate servers), continue to Step 2.

2. After the installer finishes the upgrade, start the Provisioning Database using the Service Controller.
3. Execute the installer on the DGE servers and follow on-screen prompts. During the upgrade process, the installer analyzes the aggregation scheme which is why you must start the Provisioning Database in Step 2.

   The analysis/migration process can take a large amount of time. The duration of the conversion process depends on the hardware specifications of the DGE system. For example, converting data for 15000 tests with historical data for six months:

```
1 x 3.4GHz Pentium 4; 1GBRAM; SATA drives = 3 hours
```

4. After you upgrade the DGEs, start **Traverse**. Start the server on which you installed the BVE/Provisioning Database, and then start the DGE servers. If you manually stopped any Windows services (such as the WMI service) during the installation/upgrade, start these services.

Upgrading Traverse from 5.5 or later to R92 on UNIX Platforms

1. In a distributed configuration, stop **Traverse** components on the DGE hosts, and then on the server where the BVE/Provisioning Database is operating (Step 2). In this environment, always make sure you upgrade the system running the BVE/Provisioning Database first.
2. Log in to the **Traverse** (BVE) server as root or use the `su` or `sudo` commands to obtain root permissions.
3. Shut down all **Traverse** components:

```
cd /usr/local/traverse/
etc/traverse.init stop
```

4. Back up the existing **Traverse** installation directory:

```
cd /usr/local/traverse/
utils/databaseUtil.pl --action=export
--file=/usr/local/traverse/database/provdb.xml
cd ../..
cp -r traverse OLD-traverse-5.x
This directory preserves the 5.x installation in case you abort the upgrade process.
```

5. Download and save the installation package (tar-gzipped package) to a temporary location on the server.
6. Extract the installation package and start installation by executing the following commands:

   > **Note:** (Solaris only) Use the GNU version of tar instead of the native tar utility in the following command.

```
cd /tmp
gunzip -c traverse-x.y.z-OS.tar.gz | tar xf -
cd traverse-5.6
sh install.sh
```

   If you use **Traverse** on a single server (BVE and DGE on the same host, single DGE), specify the previous **Traverse** installation directory during the installation process. The installer automatically converts configuration and historical data to version (format) R92.
7. When the installation/upgrade completes, review the configuration files from the previous version of **Traverse** in the `<TRAVERSE_HOME>/OLD-traverse-5.5` directory. If you made any modifications to the configuration files, such as adding a new JDBC driver or configuring plug-in authentication, you must manually re-apply the changes.
8. If you have **Traverse** installed on a single server only, skip to Step 11.
   If you have **Traverse** installed on more than one server (multiple DGEs, and/or BVE and DGE on separate servers), continue to Step 9.

9. After the installer finishes the upgrade, start the Provisioning Database on the server running the BVE/Provisioning Database component:

```
cd /usr/local/traverse
etc/provdb.init start
```

10. Install **Traverse** on all other servers (DGEs). You can upgrade multiple DGEs at the same time. During the upgrade process, the installer analyzes the aggregation scheme which is why you must start the Provisioning Database (Step 9).

    The analysis/migration process can take a large amount of time. The duration of the conversion process depends on the hardware specifications of the DGE system. For example, converting data for 15000 tests with historical data for six months:

```
1 x 3.4GHz Pentium 4; 1GBRAM; SATA drives = 3 hours
```

11. After you upgrade the DGEs, start **Traverse**. Start the server on which you installed the BVE/Provisioning Database, and then start the DGE servers.

```
cd /usr/local/traverse
etc/traverse.init start
```

First-time Startup

The information in this section describes how to immediately start using **Traverse** if you installed all components on a single server, and you do not have any other conflicting applications operating on the server (such as another web server or SQL database). For a distributed installation, see Traverse Configuration Files for information about the files you need to edit before using **Traverse** in a distributed environment.

The installation process creates default configuration files that enable you to operate all the **Traverse** components on the same server. The default database configuration contains:

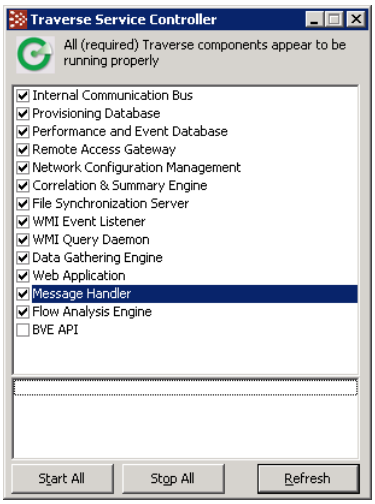
- one DGE location named `Default Location`
- one DGE component named `localhost`
- a user-class named `Default User Group`

It also creates a default user (username is `traverse`) with the password of `traverse`, and a superuser (username is `superuser`) with the password of `traverse`.

Starting Traverse for the First Time on Windows

1. On the system hosting the BVE/Provisioning Database:
   - Use the Start menu to navigate to **Traverse** programs folder.
   - Click the **Launch Traverse Service Controller** option.

- Click **Start All**.



2. Ensure that all the components display a checkmark and that a green circle displays at the top of the dialog.

   You can also open a command window and enter:

```
net start | findstr /i "traverse"
```

> **Note:** See **Starting and Stopping Traverse** *(page 26)* for more information.

3. If some components do not start, check for the following common start-up problems:
   - `Expired license key`. - Check to see if your **Traverse** license key is expired by reviewing the `<TRAVERSE_HOME>/etc/licenseKey.xml` file.
   - `Another Web server is using the httpd port on the server`.
   - `Failure to reboot after completing the installation`.
4. After identifying and fixing any problems related to component start-up, restart **Traverse**.
5. In a supported web browser, navigate to `http://your_host/`, where `your_host` is the fully qualified name or IP address of the server on which **Traverse** is operating.

> **Note:** You can specify a port number other than (the default) 80 when installing **Traverse**. Remember to include this port number in the **Traverse** URL.

6. Enter your username and password (for example, `traverse/traverse`).
7. Add some test devices to verify that the system is functioning correctly.
8. Log out of **Traverse**. Then, log in as `superuser` with the password `traverse`. See **Users and Departments** *(http://help.kaseya.com/webhelp/EN/TV/9020000/index.asp#17371.htm)* if you want to create additional departments and administration groups.
9. Populate the system with devices. See Adding Devices to add devices.

Starting Traverse for the First Time on UNIX

1. Make sure that your **Traverse** license key is not expired. (`<TRAVERSE_HOME>/etc/licenseKey.xml`).
2. Start **Traverse**. Enter:

```
cd <TRAVERSE_HOME>;
etc/traverse.init start
```

3. Make sure that all the components started and are operating correctly by executing the following command:

```
traverse.init status
```

4. See **Verifying Proper Operation** *(page 26)* for more information. Typical start-up problems include:
   - an expired license key
   - another web server is operating on the server and using the httpd port
5. After you identify and fix any problem related to **Traverse** component start-up, restart **Traverse**:

```
traverse.init restart
```

6. In a supported web browser, navigate to `http://your_host/`, where `your_host` is the fully qualified name or IP address of the server on which **Traverse** is operating.

   Note: You can specify a port number other than (the default) 80 when installing **Traverse**. Remember to include this port number in the **Traverse** URL (for example, `http://your_host:8080`).

7. Enter your username and password (for example, `traverse/traverse`).
8. Add some test devices to verify that the system is functioning correctly.
9. Log out of **Traverse**. Then, log in as `superuser` with the password `traverse`. See Users and Departments if you want to create additional departments and administration groups.
10. Populate the system with devices. See Adding Devices to add devices.

Traverse Post Discovery Tasks

After running a discovery on your network or manually adding devices, Kaseya recommends that you do the following:

- Change the password for the default user and `superuser` (Administration > **Preferences**).
- Set the correct timezone (Administration > **Preferences**).
- Specify the page to display after logging in to **Traverse**. (Administration > **Preferences**). Select a page from the **Set the page to...** drop-down menu or select **Other** and enter a specific page in the **Other** field. For example, to specify the **Manage Actions Profile** page, enter:

```
user/manageActions.jsp
```

You can obtain the URL of pages by clicking on the anchor icon in the top right-hand corner of each page. See **Show Page URL** *(http://help.kaseya.com/webhelp/EN/TV/9020000/index.asp#17077.htm)*.

- Change the DGE controller password (see **DGE Controller Port/Password** *(page 60)*).
- Update device dependencies and set up parent/child relationships if required to prevent alarm floods. See **Device Dependency** *(http://help.kaseya.com/webhelp/EN/TV/9020000/index.asp#17457.htm)*.
- Set up service containers as required to model your services. See **Service Containers** *(http://help.kaseya.com/webhelp/EN/TV/9020000/index.asp#17422.htm)*.
- Set up actions and notifications. See Actions and Notifications.
- Configure the Message Handler for monitoring traps and logs. See **Message Handler for Traps and Logs** *(http://help.kaseya.com/webhelp/EN/TV/9020000/index.asp#16774.htm)*.
- After using the system for two days, either update the thresholds manually if you are getting too many alerts, or use the "baseline" feature to automatically reset the thresholds. See Smart Thresholds Using Baselines.

IMPORTANT: If you want to use custom scripts to manage and maintain **Traverse**, you must contact Kaseya Professional Services before deploying these scripts in your **Traverse** environment.

Post-installation and Upgrade Verification

After installing or upgrading **Traverse**, make sure that the application is operating properly.

> **Note:** After an upgrade, remember to clear your browser cache.

Verifying the Traverse Installation

1. After starting **Traverse** R92 on the BVE/Web Application host system, wait 30 to 60 seconds while the application initializes.
2. In your browser, enter `http://n.n.n.n/` (where `n.n.n.n` is the IP address of the BVE/Web Application server).

> **Note:** If you changed the default port on which **Traverse** operates, make sure you enter the URL as follows: `http://n.n.n.n:port#` (where `n.n.n.n` is the IP address of the BVE/Web Application server and `port#` is the port you configured to, for example, avoid conflicts with existing web servers.

1. Log in to **Traverse** using an existing login ID and password.
2. Navigate to Status > **Devices** and make sure the severity filter is off.
3. Navigate to any device. Make sure that the date and time displayed under **Test Time** matches (or is within 5 to 15 minutes of) the current time.

   > **Note:** The DGE might require up to 30 minutes before it starts collecting metrics.

4. Navigate to any test and make sure that historical performance data displays correctly on the graphs.

New License Key

Request a New License Key

To request a license key, submit a request using **Kaseya Support** *(https://helpdesk.kaseya.com/home)* and include the following information:

- Company Name
- Service Contract ID
- Number of devices and tests

Installing A New License Key

Traverse components use a license key that determines the features available for use. When and if the license key expires, **Traverse** ceases operation.You will need to acquire and install a new key from **Kaseya Support** *(https://helpdesk.kaseya.com/home)*.

You need to install a new key if the key is temporary (for trial purposes) and expires, or if the license key format changes between versions of **Traverse**.

> IMPORTANT: If you are running **Traverse** v5.5 or later, you do not need to restart **Traverse** if you want to install a new license key with a new expiration date or new number of devices and DGEs. Just copy your new license key into the `TRAVERSE_HOME/etc` directory and if **Traverse** is already running, it will automatically reload the configuration file within an hour.

Installing a New License Key on Windows

1. Save or copy the `licenseKey.xml` file to `<TRAVERSE_HOME>\etc\`.
2. Make sure to replace the existing `licenseKey.xml` file (if any).
3. Restart **Traverse**:
   - Start > Programs > Traverse > **Stop Traverse.Components**.
   - Then, Start > Programs > Traverse > **Start Traverse Components**.

Installing a New License Key on UNIX

1. Save or copy the `licenseKey.xml` file to `<TRAVERSE_HOME>/etc/`.
2. Make sure to replace the existing `licenseKey.xml` file (if any).
3. Restart Traverse:

```
<TRAVERSE_HOME>/etc/traverse.init restart
```

Windows System Performance Tuning

IMPORTANT: Make sure you back up configuration files before you make any changes to these files.

Name Servers

You can increase system performance by deploying a caching name server on the servers on which the DGE components operate.

Increasing Java Virtual Memory (JVM) Size

The DGE, BVE ObjectStore, and Web Application operate as separate processes and have their own Java Virtual Memory settings. If you add additional RAM on servers hosting DGEs, Kaseya recommends that you increase the JVM size that the DGEs use.

Increasing the JVM Size

- Shut down the DGE. Navigate to Start > Control Panel > Admin Tools > Services. Right-click **Traverse Data Gathering Engine** and select **Stop**.
- Using a text editor with word wrapping disabled, add the following line to the end of the `<TRAVERSE_HOME>\bin\monitor.lax` file as follows:

```
lax.nl.java.option.additional=-xmx512m
```

- This allocates 512MB of memory for the DGE process.
- Save and close the `monitor.lax` file.

Make sure you always dedicate physical memory (RAM) to the java process, and not swap space. For example, if you have 2GB of swap space, but only 512MB of RAM, set the JVM size to less than 512MB (do not set it to 2GB).

System Security

Kaseya strongly recommends that you terminate or disable all unnecessary services and processes on **Traverse** servers (this includes TELNET and FTP).

Internet Explorer Browser Settings

Make sure that you enable the following settings in Internet Explorer (Tools > Internet Options > Security > **Custom Level**):

- Binary and Script behaviors
- Script ActiveX Controls marked Safe for scripting
- Allow Scripting of Internet Explorer Web Browser Controls
- Allow Script-initiated windows without size or position constraints
- Scripting > Active Scripting
- Scripting > Allow Paste Operations Via Script
- Scripting > Scripting of Java Applets

UNIX System Performance Tuning

> **IMPORTANT:** Make sure you back up configuration files before you make any changes to configuration files.

Name Servers

You can increase system performance by deploying a caching name server on the servers on which the DGE components operate.

On Solaris platforms, you can run NCSD with the following parameters:

```
positive-time-to-live hosts 10800
keep-hot-count hosts 200
check-files hosts no
check-files ipnodes no
check-files exec_attr no
check_files prof_attr no
check_files user_attr no
```

Disk I/O

If you are using IDE drives, you can increase I/O performance by enabling 32bit I/O, direct memory access and multi-block reads by entering:

```
hdparm -c1 -d1 -m16 /dev/hda
```

Make sure to replace `/dev/hda` with the correct device name appropriate for your system. Add this command to `/etc/rc.local`.

Increasing the File Descriptors

1. Increase the file descriptors to 8192 by adding the following parameters to the `/etc/security/limits.conf` file.

```
* soft nofile 8192
* hard nofile 8192
```

2. Edit the `/etc/pam.d/login` file, and add the following:

```
session required /lib/security/pam_limits.so
```

3. Increase the system-wide file descriptor limit by adding the following three lines to the `/etc/rc.d/rc.local` startup script:

```
# Increase system-wide file descriptor limit.
echo 4096 > /proc/sys/fs/file-max
echo 16384 > /proc/sys/fs/inode-max
```

Increasing Java Virtual Memory (JVM) Size

The DGE, BVE ObjectStore, and Web Application operate as separate processes and have their own Java Virtual Memory settings. If you add additional RAM on servers hosting DGEs, Kaseya recommends that you increase the JVM size that the DGEs use.

Increasing the JVM Size

1. Shut down the DGE.
   `<TRAVERSE_HOME>/etc/monitor.init` stop
2. Edit `<TRAVERSE_HOME>/etc/monitor.init` and search for `Xmx1024`. Replace this value with `Xmx1536`. This adds an additional 512MB of memory for the DGE process.
3. Save and close the `monitor.init` file.

   Make sure you always dedicate physical memory (RAM) to the java process, and not swap space. For example, if you have 2GB of swap space, but only 512MB of RAM, set the JVM size to less

than 512MB (do not set it to 2GB).

System Security Issues

Kaseya strongly recommends that you terminate or disable all unnecessary daemons and processes on the **Traverse** servers. This includes telnet and ftp.

Use ssh if you need to log in to the **Traverse** server and scp for all file transfers. See **Using Traverse Behind Firewalls** *(page 21)* for more information about firewalls.

Configuring SSL for the Web Application

Since the **Traverse** Web Application is pure HTML based, the GUI component can be accessed using both regular and secure (SSL) HTTP protocol. By default SSL is already enabled on the default port 443 with a Kaseya certificate, but to enable or change the certificate for SSL, use the following steps:

> **Note:** These changes will need to be re-applied when you install a new version of **Traverse**.

Configuring SSL for the Web Application

1. The application server (Apache Tomcat) used by **Traverse** uses a JKS format keystore. **Traverse** by default ships with a keystore with a self-signed certificate. If you are not ready to install a valid key yet, you can skip to Step 10. Otherwise, first rename or move the existing keystore located at `<TRAVERSE_HOME>/plugin/web/webapp.keystore`
2. Create a private/public (RSA) key pair using the following command:

```
<TRAVERSE_HOME>/apps/jre/bin/keytool -genkey -keyalg RSA -storepass changeit -alias
tomcat -keystore <TRAVERSE_HOME>/plugin/web/webapp.keystore
```

3. Answer the questions, making sure to specify the fully-qualified domain name when asked for first/last name. Do not use comma (,) in any of the answers as it will cause problems. When asked for key password for tomcat, press return/enter.
4. Generate a Certificate Signing Request (CSR) using the following command:

```
<TRAVERSE_HOME>/apps/jre/bin/keytool -certreq -storepass changeit -alias tomcat
-keystore <TRAVERSE_HOME>/plugin/web/webapp.keystore -file my_new_key.csr
```

5. You will need to send the CSR (`my_new_key.csr`) to a valid certificate authority (CA) such as Verisign or Thawte. Usually the CA will send you a signed certificate via email. If you are acting as your own CA, the CSR can be signed using OpenSSL or other SSL tools.
6. Save the certificate in `my_new_cert.pem` and make sure that the certificate begins with -----BEGIN CERTIFICATE----- and ends with -----END CERTIFICATE-----. All other text above/below the specified section should be deleted.
7. Import the new certificate into a new keystore using:

```
<TRAVERSE_HOME>/apps/jre/bin/keytool -import -v -trustcacerts -alias tomcat
-storepass changeit -file my_new_cert.pem -keystore
<TRAVERSE_HOME>/plugin/web/webapp.keystore
```

8. When asked **Trust this certificate?**, answer yes and the certificate will be installed into the keystore.
9. Verify that the certificate has been imported correctly using:

```
<TRAVERSE_HOME>/apps/jre/bin/keytool -list -v -storepass changeit -keystore
<TRAVERSE_HOME>/plugin/web/webapp.keystore
```

10. Edit `<TRAVERSE_HOME>/apps/tomcat/conf/server.xml` using a text editor and check that the following section is uncommented and not enclosed between (`<!-- .. -->`) :

```
<Connector port="443"
  minProcessors="20" maxProcessors="80"
  enableLookups="false" allowChunking="false"
  acceptCount="100" scheme="https" secure="true"
  clientAuth="false" sslProtocol="TLS"
  keystorePass="mypassword"
  keystoreFile="conf/.keystore"
  compression="off" debug="0"
  URIEncoding="UTF-8" />
```

11. Make sure that the `keystore`, `keystorepass` and `port` parameters are set correctly. On a Windows platform, the path would be specified as `/C:/Program Files (x86)/Traverse/plugin/web/webapp.keystore` in this file.
12. To configure Tomcat to use only SSL (https), you can disable the standard http request handler as described below.
13. Save the file and restart the Web Application if already running. On Linux or Solaris hosts:

```
<TRAVERSE_HOME>/etc/webapp.init restart
```

14. On Windows hosts, click **Launch Traverse Service Controller** from the Windows Start menu to display the **Traverse Service Controller**. First clear the Web Application check box and click **Apply** to stop the Web Application. Then wait 15-30 seconds, select the Web Application check box and click **Apply** to start the Web Application.
15. Wait 15-30 seconds for the Web Application to initialize and use your web browser to connect to `https://your_traversetraverse_host/` and you should see the **Traverse** login page.

Disabling non-SSL Web Application server

If you want to use only SSL, you can disable the non-SSL server of the Web Application by performing the following steps:

1. Edit `<TRAVERSE_HOME>/apps/tomcat/conf/server.xml` using a text editor and locate the following Connector section for port 80:

```
<!-- define standard http request handler -->
<Connector port="80" minProcessors="20" maxProcessors="80" enableLookups="false"
allowChunking="false" acceptCount="100" redirectPort="443" compression="off"
debug="0" URIEncoding="UTF-8" />
```

2. Comment out the section by adding "<!--" and "-->" as follows:

```
<!-- define standard http request handler -->
<!-- disabled
<Connector port="80" minProcessors="20" maxProcessors="80" enableLookups="false"
allowChunking="false" acceptCount="100" redirectPort="443" compression="off"
debug="0" URIEncoding="UTF-8" />
-->
```

3. Save the file and restart the Web Application if already running. On non-Windows hosts:

```
<TRAVERSE_HOME>/etc/webapp.init restart
```

On Windows hosts, click **Launch Traverse Service Controller** from the Windows Start menu to display the **Traverse Service Controller**. First clear the Web Application check box and click **Apply** to stop the Web Application. Then wait 15-30 seconds, select the Web Application check box and click **Apply** to start the Web Application.

The Web Application should now be accessible only via the `https://your_traverse_host/` URL and not http (plain text).

Redirecting non-SSL Port to SSL Port Automatically

Edit `<TRAVERSE_HOME>/webapp/WEB-INF/web.xml` and add the following block of data immediately after the opening <web-app> tag structure:

```
<!-- This block forces SSL for all connections  -->
<security-constraint>
  <web-resource-collection>
  <web-resource-name>Entire Application</web-resource-name>
  <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>  <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

Now restart the Web Application to activate the new settings.

Using Traverse Behind Firewalls

If any component of **Traverse** is going to be installed behind a firewall, depending on the existing policies, some changes may be necessary to the rules to accommodate the requirements. In the following requirements, "remote" host implies a host that is outside of the firewall while a "local" host is a device on the secure side of the firewall. Also, note that the requirements are not applicable for cases where the two hosts in question are on the same side of the firewall (i.e. packets are not crossing the firewall).

Requirements for the BVE Provisioning Database

The provisioning server stores all device, test, action, threshold, authentication and other provisioning information. This information is retrieved on-demand by both the web servers and DGEs. This is accomplished by creating connections to the database server on specific TCP ports running on the provisioning host. The following firewall rules will need to be applied for a provisioning server which is behind a firewall:

Firewall Rules for a Provisioning Server that is Behind a Firewall

Protocol	Direction	Local Port	Remote Host	Remote Port	Reason
tcp	incoming	7651	any	any	Traverse Provisioning Database
tcp	incoming	7652	any	any	Traverse Provisioning Database
tcp	incoming	7653	any	any	Traverse messaging protocol #1
tcp	incoming	7654	any	any	Traverse messaging protocol #2
tcp	incoming	7661	any	any	Traverse BVE (provisioning) API server
udp	incoming	162	any	any	snmp traps
tcp	outgoing	any	any DGE	7657	external data feed API server
tcp	outgoing	any	any DGE	7659	input stream monitor
udp	outgoing	any	DNS servers	53	DNS queries for name resolution

Requirements for Web Servers

The web servers provide an interface for displaying all collected information as well as reports generated from those information. If a location is served by more than one web server, a load balancer is installed to distribute the load and the load balancer will need the same firewall rule changes as the web servers themselves. The load balancer might have additional firewall specific requirements. You must apply the following firewall rules for web servers which are behind a firewall:

Firewall Rules for a Web Server that is Behind a Firewall

Protocol	Direction	Local Port	Remote Host	Remote Port	Reason
tcp	incoming	80	any	any	any access to Web Application
tcp	incoming	443	any	any	any access to Web Application over ssl
udp	outgoing	any	DNS servers	53	DNS queries for name resolution

Requirements for DGE (monitors)

The DGEs perform actual monitoring of all provisioned devices and store the data on a local database. The web servers will need access to this stored data on-demand for report generation. The provisioning server also needs access to the data to fulfill requests made via the BVE socket API. Since the DGE perform monitoring tasks, it will need outbound access via a multitude of ports and protocols. The following firewall rules will need to be applied for a DGE server which is behind a firewall:

Firewall Rules for a DGE that is Behind a Firewall

Protocol	Direction	Local Port	Remote Host	Remote Port	Reason
tcp	incoming	7657	any	any	external data feed API server
tcp	incoming	7659	any	any	input stream monitor
tcp	incoming	7663	web app	any	DGE database lookup
tcp	incoming	7655	any	any	DGE status server
tcp	incoming	9443	dge-extensions	any	from DGE-extension to upstream DGE
tcp	outgoing	any	WMI query server	7667	dge connection to WMI query server
tcp	incoming	20	any	any	FTP servers create incoming connection on port 20 in response to connections on port 21
icmp	outgoing	any	any	"echo"	packet loss, round trip time tests
udp	outgoing	any	any	161	SNMP queries
udp	outgoing	any	any	53	DNS queries, tests
udp	outgoing	any	any	123	NTP service tests
udp	outgoing	any	any	1645	radius service tests
tcp	outgoing	any	any	21	FTP service tests
tcp	outgoing	any	any	25	SMTP service tests, alerts via email
tcp	outgoing	any	any	80	HTTP service tests
tcp	outgoing	any	any	110	POP3 service tests
tcp	outgoing	any	any	143	IMAP service tests
tcp	outgoing	any	any	389	LDAP service tests
tcp	outgoing	any	any	443	HTTP over ssl service tests
tcp	outgoing	any	any	993	POP3 over ssl service tests
tcp	outgoing	any	any	995	IMAP over SSL service tests
tcp	outgoing	any	windows	135	WMI queries to Windows hosts being monitored via DCOM. See Apache Web Monitor.

Firewall ports for DGE-extensions

The DGE-extensions make all outbound connections to an upstream DGE and BVE, and there are no incoming connections to the DGE-extension. The following TCP ports need to be opened on the upstream DGE location to all the DGE-extensions to connect:

Port	From	To
TCP/7651	DGE-x	BVE
TCP/7652	DGE-x	BVE
TCP/7653	DGE-x	BVE
TCP/7654	DGE-x	BVE
TCP/9443	DGE-x	DGE

Using Traverse in NAT Networks

NAT (Network Address Translation) devices usually translate connections between a public network and a private address space. There are several issues to consider while monitoring in a NAT network:

- **NAT Port Translation**: In this NAT method, one or more public IP address are mapped to one or more private IP addresses by manipulation of the source port. It is difficult to permit an external monitoring server to query an internal host unless such translation is set up.
- **Firewalls Disable Queries from public network**: Several NAT and firewall devices (such as the PIX firewall) disable SNMP queries from their public interfaces.
- **Dynamic NAT**: For non-server type devices (such as user systems), they usually get a dynamic IP address instead of a fixed address. These devices cannot be queried since the IP address is changing all the time.

Traverse can be deployed in a NAT environment as long as there is a way to query the device being monitored. If the DGE is co-located near the private LAN, then an ethernet interface from the DGE can be attached to the NAT network directly.

Traverse can be deployed in an environment with similar private addresses, as long as each of these networks has its own DGE. The Provisioning Database does NOT reference devices by IP addresses, so many devices can exist in the system with the same IP address. Each device is allocated to a DGE, so as long as the respective DGE can access the private (or NAT) network, the devices on these networks can be monitored by **Traverse**.

Adding an Additional DGE

You can add additional DGEs in order to increase the scalability of your **Traverse** installation. You might need to purchase a license in order to have more than one. The steps to do this are described in **Configuring DGEs** *(page 30)*.

High Availability Configurations

The **Traverse** distributed database and processing architecture allows very high levels of fault tolerance and scalability during deployment. All of the components in the various tiers are horizontally scalable which is essential for expansion and real-time performance reports.

All of the configuration information is stored in the BVE Provisioning Database. On startup, the DGEs connect to the BVE Provisioning Database and download a local copy of their configuration. Any

updates made to the BVE Provisioning Database are pushed out in real time to the corresponding DGE.

To handle the case of a DGE physical server going down, you can set up a spare 'hot standby' server in any central location (N+1 redundancy) which has the software installed and configured. In the case of a production DGE going down for an extended period of time due to hardware failure, you can set the name of the DGE in the `dge.xml` configuration file (see **DGE Identity** *(page 60)*) and start **Traverse** on the backup server. This backup DGE automatically connects to the BVE Provisioning Database and downloads the configuration of the failed DGE. When the production DGE comes back up, it can be even run in parallel before shutting down the backup DGE. The only caveat is that the performance data collected during this interval will be missing on the production DGE.

If desired, you can have a backup DGE for each of the production DGEs (N+N redundancy) but this is not really needed if the centralized DGE can poll all the data remotely.

If connectivity between the DGE and the BVE database is lost, the DGE continues to poll, aggregate and even generate alarms completely independently. When connectivity to the BVE database is restored, the DGE restarts and downloads a fresh copy of its Provisioning Database.

The BVE database can be replicated on multiple servers for fault tolerance.

The performance database which is local to each DGE can be located on a remote database cluster if needed for fault tolerance also. The JDBC communication between the DGE and the performance database allows such a setup seamlessly just by a few configuration file changes. Contact Kaseya Professional Services for information and pricing for this configuration service.

Lastly, the Web Application and reporting engine also gets all the configuration information from the BVE database server on startup and hence you can have any number of Web Application servers behind a load balancer for fault tolerance as well as distributed report processing.

Chapter 3

Starting Traverse

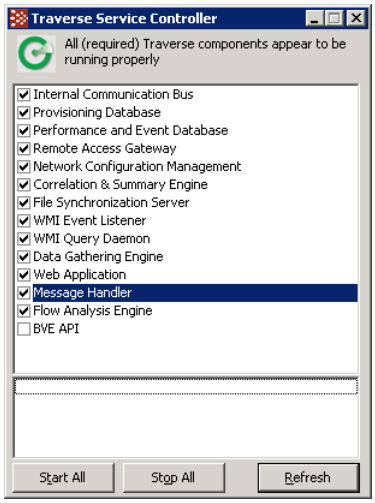
In This Chapter

Starting and Stopping Traverse - Windows

Windows

On the system hosting the BVE/Provisioning Database:

1. Use the Start menu to navigate to **Traverse** programs folder.
2. Click the **Launch Traverse Service Controller** option.
3. Click **Start All**.



You can also start and stop **Traverse** components by opening a windows command prompt and using:

`net start service_name` and `net stop service_name`

where `service_name` is the Windows Service listed in the following table.

To Automatically Start Services on Reboot

To control the startup of individual components, use the Service Control Manager from the Windows menu: Control Panel > Administrative Tools > **Services**. All **Traverse** service names are prefixed with `Traverse`. If you want **Traverse** components to start when the system starts, select all or individual **Traverse** services and change the Start-up type to `Automatic`. You can also do this using the command prompt and entering:

```
sc config tvSlaMgr start=auto
```

If you are operating the Web Application and DGE monitor components on the same host, set the start-up properties for these services to `Disabled`.

Traverse Windows Services

Windows Service	Description	Default
`nvprovdb`	BVE/provisioning server database (poet)	
`nvdgedb`	DGE/monitor database (mysql)	
`nvmonitor`	DGE/monitors	
`nvwebapp`	web interface	
`nvbveapi`	BVE API server	Disabled
`nvmsgsvr`	Message Handler	
`tvSlaMgr`	SLA Manager	Disabled

To stop **Traverse**, from the Windows menu: Start > Programs > Traverse > **Stop Traverse Components**.

> **Note:** If you have recently stopped the Provisioning Database, it may take a few seconds until you can start the database again while it shuts down completely. The startup scripts will let you know if the Poet database was unable to start up properly and you should try again after a few seconds.

Verifying Proper Operation

Check the status of individual components using the Service Control Manager. The **Status** column displays `Started` when a **Traverse** component is operating. You can also execute the following command from a command prompt to get a list of all operating services:

```
net start | more
```

Traverse components are prefixed with `Traverse` in the display.

Starting and Stopping Traverse - UNIX

Traverse components are started and stopped using the `<TRAVERSE_HOME>/etc/traverse.init` script. You should execute this script with the start parameter from `/etc/rc.local` or another startup directory relevant to your operating system. This enables **Traverse** components to start automatically when the system starts.

Before you can use the script, you must edit the script and uncomment the components you want to operate on the server. For example, if you are operating the Web Application and DGE monitor components on the same host, edit `traverse.init` as follows:

```
PROVDB="N"
BVEAPI="N"
WEBAPP="Y"
MESSAGE="Y"
DGE="Y"
SLAMGR="Y"
```

Each **Traverse** component has its own startup script. This allows you to start and stop individual components. The scripts are in the `<TRAVERSE_HOME>/etc` directory and are described in the following table:

Traverse Service Start/Stop Scripts

Script Name	Description
`provdb.init`	BVE/provisioning server database (poet)
`dgedb.init`	DGE/monitor database (mysql)
`monitor.init`	DGE/monitors
`webapp.init`	web interface
`bveapi.init`	BVE API server
`msgsvr.init`	Message Handler
`slamgr.init`	SLA Manager

Each of these scripts starts and stops with the `start` and `stop` command line option.

To start **Traverse**, execute the following command:

```
sh# /etc/init.d/traverse.init start
```

If you start **Traverse** by starting individual services, make sure you start the Provisioning Database first. This is because all other **Traverse** components request configuration information from the Provisioning Database during startup.

Start the DGE database and monitors after the Provisioning Database. They provide the status of all configured devices and tests. Then, start the Web Application, followed by the BVE socket server.

To stop **Traverse**, execute the following command:

```
% <TRAVERSE_HOME>/etc/traverse.init stop
```

When shutting down **Traverse** by shutting down individual components, make sure you shut down the components in the opposite order they are required to be started as described above.

If you want to stop the components of **Traverse** that read configuration files (so that they can read the configuration files again), execute the following command:

```
% <TRAVERSE_HOME>/etc/traverse.init stopcore
```

This command does not stop the databases or the messaging bus.

> **Note:** After you shut down the Provisioning Database, wait at least 10 to 20 seconds before attempting to start Provisioning Database. If you attempt to restart the Provisioning Database too soon, the startup scripts inform you that the Poet database is unable to start-up properly.

Verifying Proper Operation

Use the status parameter with the `traverse.init` script to display the status of the different components. For example:

```
./traverse.init status
messaging server (openjms)      ... running
provisioning database (poet)    ... running
independent message handler     ... running
dge (monitor) components        ... running
dge/jms database (mysql)        ... running
application server (tomcat)     ... running
```

Alternatively, you can use status parameter with other startup scripts to check the status of individual components.

Logging In

Traverse users in the `superusers` admin-group can log in using the procedure below. If you are not a **Traverse** superuser, `superuser` or your administrator must create the admin-group structure and assign you to an admin-class which determines your permissions (to view, create, modify, and delete entities within the application.

Before you log in, you need to have received a username and password from `superuser` or your administrator.

Logging in to Traverse

1. Type `http://traverse.your.domain` into your web browser.

   > **Note:** If you configure an alternate port number other than the default (port 80), remember to include this port number in the **Traverse** URL. See **Web Server TCP/IP Port** *(page 62)* for more information on configuring the web server port.

2. Enter your **Username** and **Password**, and then click **Login**.
3. To have your password emailed to you, click **Forgot your password? Click here**.
4. Click **Login** to enter the site.

Chapter 4

DGE Management

The following topics provide a detailed explanation for on premise users of how to manage DGEs and DGE extensions. Traverse cloud users are provided separate instructions for downloading and installing DGE extensions.

In This Chapter

Overview

Traverse uses a distributed, tiered architecture where the data collection and storage is handled by the Data Gathering Engine (DGE) component. Each DGE polls data from the network devices, servers and applications and performs real-time aggregation and storage of this performance data in a local relational database. The DGE also triggers actions and notifications when it detects that the threshold conditions are exceeded or crossed. Each DGE also processes its data during report generation, which allows for parallel and distributed processing for very large environments.

A DGE-extension is a light-weight remote monitoring and collection component which has no local storage. It pushes all the data it collects to a DGE over a secure SSL "push" connection which makes it ideal for deploying behind firewalls for monitoring small NAT environments.

Configuring DGEs

If you would like to expand your **Traverse** system to monitor additional devices in remote geographical or logical locations, you can install a Data Gathering Engine (DGE) on another physical machine and integrate it into your existing setup. You can add multiple DGEs in the same location for load balancing or increasing monitoring capacity.

Adding a Location

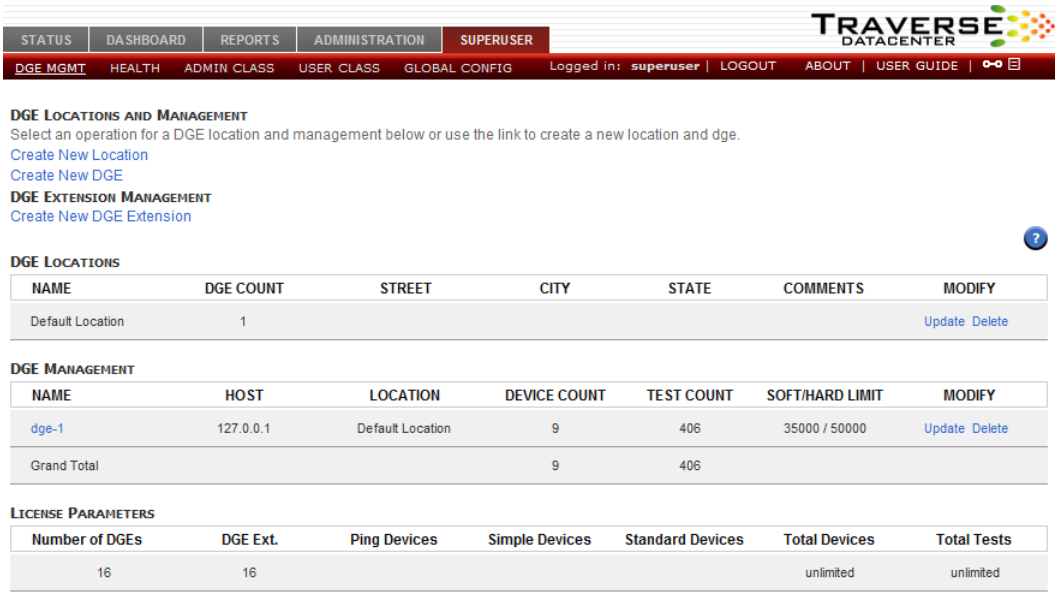
Note: Creating a new location is optional. You must assign a new DGE to a location during configuration. You can add it to a new location or an existing location.

Locations

DGEs are grouped within DGE locations. A DGE location is simply a way of grouping DGEs for load balancing; The location can be any logical or functional name, for example, *New York*, *datacenter3*, *finance*. DGEs in the same DGE location need not be in the same physical location.

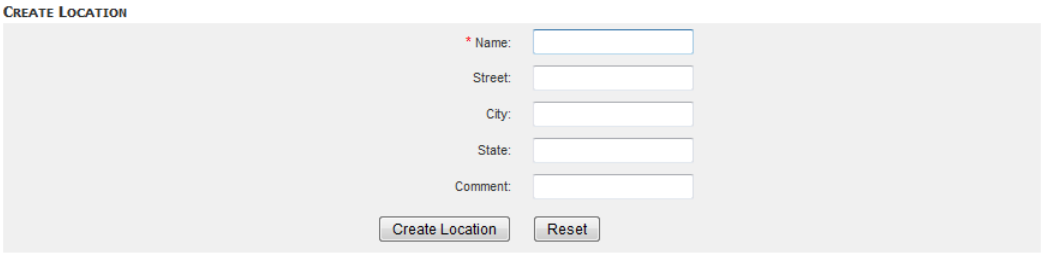
Load Balancing, Hard Limits, Soft Limits

For multiple DGEs in a single "location", **Traverse** uses a load balancing mechanism based on configurable test limits to ensure that DGE hosts are not overloaded. There are two limits, soft and hard, which are used to determine whether the DGE has the capacity to take on a newly-provisioned device. If the number of tests reach the hard limit, no more tests can be provisioned on that DGE. Once a soft limit is reached, only tests for existing devices can be added to that DGE. Else the device is provisioned on the least loaded DGE. Note that tests for a device are not split across multiple DGEs to optimize performance.



Creating a DGE Location

1. Log in to the **Traverse** Web Application as `superuser`.
2. Navigate to Superuser > **DGE Management**.
3. Click **Create New Location**.
4. Fill in the **Name** field with a unique name to identify the DGE host location (required). This can be any text, typically the name of a geographic location, department, building, etc.
5. Fill in the optional fields if desired to clarify the geographical location of the DGE host and any comments.
6. Click **Create Location** to save your changes.
7. Repeat steps 3-6 above as needed to create additional locations.



Adding a DGE

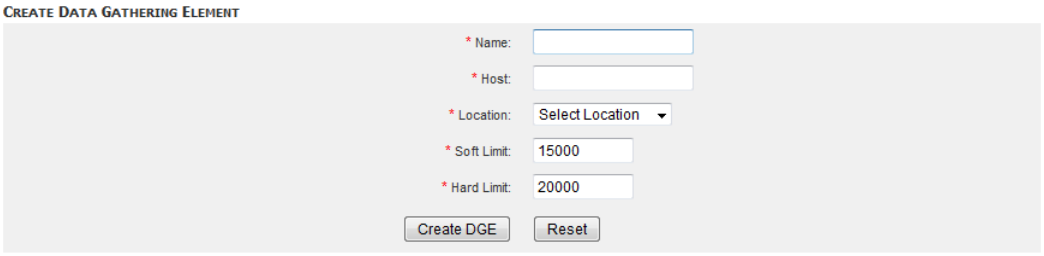
Prerequisites

- Ensure that the **Traverse** Business Visibility Engine (BVE) component is installed and operating correctly.

- Identify the host name and IP address of the system that will host the DGE.
- Review **Using an Existing MySQL Database with a DGE** *(page 34)* if you intend on using an existing MySQL database with your new DGE.
- Review **Disk Space Requirements for DGE Aggregation** *(page 34)* for guidelines on sizing the disk space for your new DGE.

Create a DGE Record in Traverse

1. Log in to the **Traverse** Web Application as `superuser`.
2. Navigate to Superuser > **DGE Management**.
3. Click **Create a New DGE**.



4. Fill in the **Name** field with a unique name to identify the DGE host. This name can be arbitrary, but should be unique as it is used by the DGE to identify itself to the BVE layer.
5. Fill in the **Host** field with the fully qualified domain name or IP address of the DGE host. At startup, the DGE verifies the hostname/IP in the Provisioning Database against its own IP address.
6. Select the **DGE location** from the drop-down list. There may often be multiple DGEs assigned to a single geographic location.
7. Set the **Soft Limit** and **Hard Limit** values. Accept the default values if you don't have a reason to change them. See *Load Balancing, Soft Limits, Hard Limits* in **Adding a Location** *(page 30)* for more information.
8. Click **Create DGE** to save your changes.
9. Repeat steps 3-8 above as needed to create additional DGE hosts.

Installing the DGE

1. Locate the same installer you used to install the BVE component, appropriate for Windows or UNIX.
2. Install the DGE software on a new DGE host and reboot the system.

3. Subsequent screens ask you to specify:

- The **IP Address** of the *Fully Qualified Domain Name (FQDN) or IP address* of the system hosting the BVE/Provisioning Database.
- The unique **DGE Name** of your new DGE. This should match the **Name** you entered in step 4 of the *Create a DGE Record in Traverse* procedure above.
- The email address and SMTP mail server your new DGE will use to notify you.

After the Installation

1. Start the DGE if it is not already started.
2. Check the health of the DGE you have just installed by navigating to the SuperUser > **Health** page. Your DGE should be listed and show all components with an OK icon. See **Monitoring DGE Operation and Capacity** *(page 38)* for more information.

- Log into the Traverse Web Application and run Network Discovery to add devices and tests to the new DGE.

Troubleshooting

If you encounter connection issues, verify on the system hosting the new DGE:

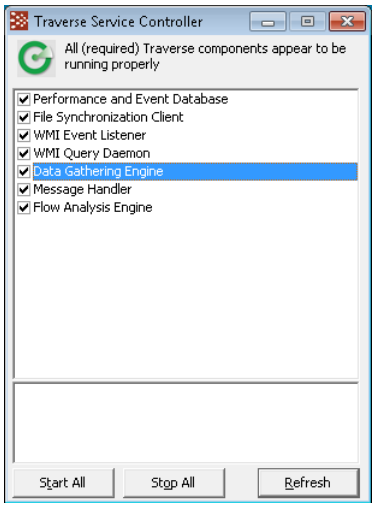
- The *name* of the DGE in the `<TRAVERSE_HOME>/etc/dge.xml` file. Locate the string `<dge name`. It should match the **Name** specified on the Superuser > **DGE Management** page.
- The *Fully Qualified Domain Name (FQDN) or IP address* of the system hosting the BVE/Provisioning Database. The host value is located in the `<TRAVERSE_HOME>/etc/emerald.xml` file. Locate multiple instances of the string `host`.

Restarting the DGE

If you make changes to either XML, the DGE service must be restarted. On the system hosting the DGE:

For Windows:

1. Use the Start menu to navigate to **Traverse** programs folder.
2. Click the **Launch Travese Service Controller** option.
3. Uncheck the **Data Gathering Engine** component
4. Check the **Data Gathering Engine** component to restart it.



For UNIX

- Run the following: `<TRAVERSE_HOME>/etc/dge.init`

Using an Existing MySQL Database with a DGE

By default, the DGE database is set to MySQL, which is licensed and shipped with **Traverse**.

If you have a copy of MySQL already running on the host where the DGE component of **Traverse** will be installed, it is strongly recommend that the bundled version of MySQL for the DGE database should be used. This copy of MySQL has been tuned for optimal performance, and some of these settings might not be compatible with your existing installation/databases. Also, the existing instance of MySQL may be incompatible with the database drivers we are using. If there is already a copy of MySQL installed on the **Traverse** host, you can run the MySQL bundled with **Traverse** on a different port to avoid conflict.

Disk Space Requirements for DGE Aggregation

Note: A DGE Disk Space Requirements calculator is available from **Kaseya Support** *(https://helpdesk.kaseya.com/home).*

The DGE database stores three main data types:

- Aggregated performance data
- Event data (threshold violations)
- Syslog and Trap text messages

Each aggregated data value is 30 bytes in size (including the size of its index). For the default aggregation scheme:

```
5 minute samples for 1 day = 60/5*24 = 288 samples
15 minute samples for 7 days = 60/15*24*7 = 672 samples
60 minute samples for 90 days = 60/60*24*90 = 2160 samples
1 day samples for 3 years = 1*365*3 = 1095 samples
TOTAL size per test = (288+672+2160+1095) * 30 bytes = 126 KB per test
For 10,000 tests DGE database = 1.26GB
```

The database size for 10,000 tests, using some alternate aggregation schemes, are described in the table below.

Database Size for Specific Aggregation Schemes

Aggregation Scheme	DB Size for 10,000 Tests
5 min for 1 day, 15 min for 1 week, 1 hour for 3 months, 1 day for 3 years	1.3GB
5 min for 1 day, 15 min for 1 week, 1 hour for 1 month, 1 day for 2 years	0.75GB
5 min for 1 day, 15 min for 1 month, 1 hour for 3 months, 1 day for 2 years	1.8GB
5 min for 1 day, 15 min for 1 week, 1 hour for 6 months, 1 day for 2 years	1.9GB
5 min for 30 days, 30 min for 3 months, 2 hours for 6 months, 1 day for 3 years	4.8GB

Updating an Existing DGE

1. Log in as `superuser`.
2. Navigate to Superuser > DGE Management > **Update**.
3. Enter the new **Name**, **Host** (IP address), **Soft Limit** or **Hard Limit**.
4. Click **Update DGE**.

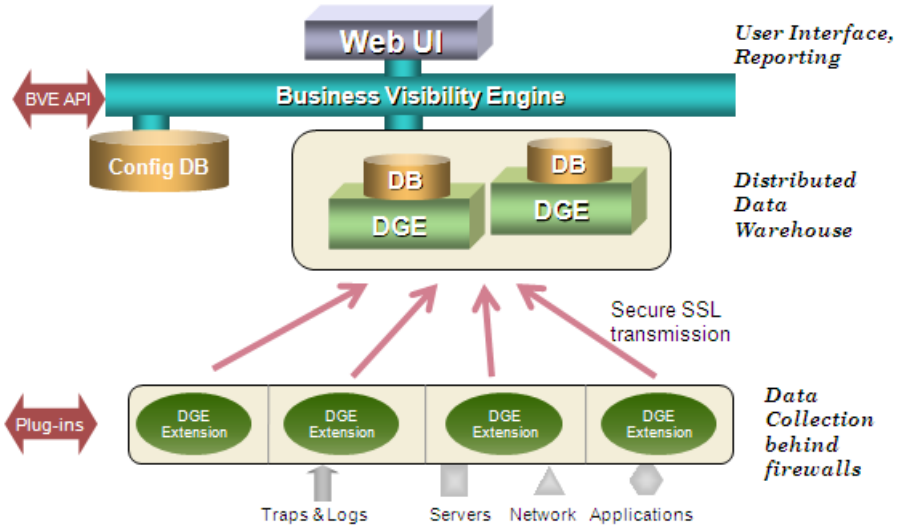
Configuring DGE Extensions

A DGE-extension is a light-weight component designed to extend the data collection of a DGE over a network. The DGE-extension collects data, but does not store it locally. The collected data is periodically transferred to an upstream DGE for aggregation and storage. Using a DGE-extension can help to minimize data collection network traffic between subnets or different geographic locations without installing additional DGEs.

Additionally, DGE-extensions initiate the connections to the BVE Provisioning Database and the DGE, so they do not require a publicly accessible IP address. The data transfer to the DGE is done over an SSL tunnel.

Traverse System Components Including DGE-extensions

The following figure shows how DGE-extensions fit into the **Traverse** system architecture.



The steps to configure a DGE-extension are generally the same as for a regular DGE, but each DGE-extension is tied to an upstream DGE for its data storage:

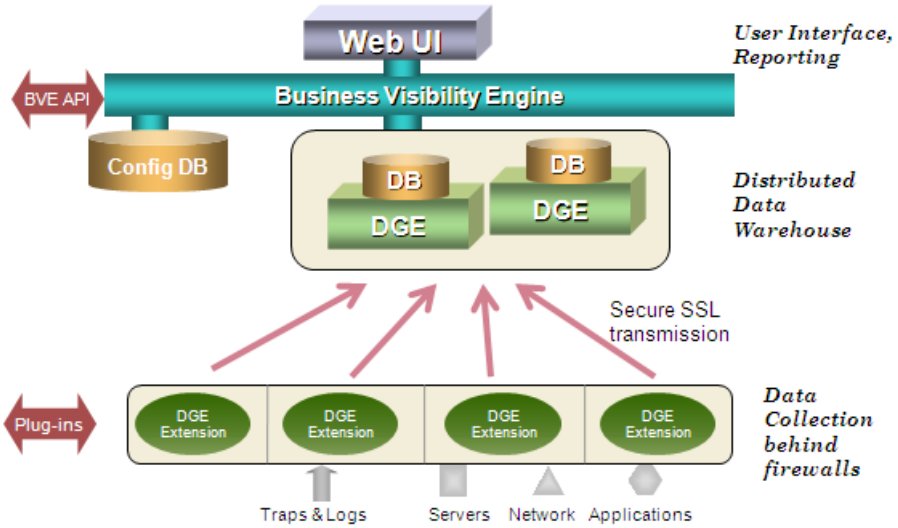
- First, configure the DGE-extension itself with a unique name (`etc/dge.xml`), just like with a DGE
- Then, log in to the Web Application as `superuser` and add the DGE-extension name as a new DGE-extension.

Adding a DGE Extension

1. Log in to the **Traverse** Web Application as `superuser`.
2. Navigate to Superuser > DGE Mgmt > **Create New DGE Extension**.
3. In the **Create DGE Extension** form, configure the properties:
   - **Unique Name**: The same unique name as configured on the DGE-extension host.
   - **Description**: Provide some additional descriptive information.
   - **Upstream DGE**: From the drop-menu menu, choose the real DGE that you want the DGE-extension to send data to.
   - **Upstream DGE Fully Qualified Host Name / IP Address**: Specify the IP address or hostname of the upstream DGE. On a DGE behind a NAT address space, you must enter an IP address for the DGE which is reachable by the DGE-extension. Do NOT enter localhost or 127.0.0.1 or some internal unreachable IP address since the DGE-extension will try to connect to this IP address.

> **Note:** By default, the configured address of the DGE is filled in. If the upstream DGE has multiple IP addresses, set this field to an IP address that is reachable by the DGE-extension.

- ✓ **Soft Limit**
- ✓ **Hard Limit**



4. Click **Create DGE Extension**.
   The new DGE-extension appears under **DGE Management**, and it will be available as a location option when users add devices.

Configuring the DGE Name

1. Install the DGE software on the new DGE-extension host and reboot the system.
2. On the DGE-extension host, edit `<TRAVERSE_HOME>/etc/dge.xml` to verify the unique name of this DGE-extension. The install process will automatically set this for you, so you will only need to edit this file if you are making name changes after installation.
3. Log in to the BVE as an end user and add devices and tests to the DGE.

Updating a DGE Extension

Modifying the Properties of a DGE-extension

1. Log in to the **Traverse** Web Application as `superuser`.
2. Navigate to Superuser > **DGE Mgmt**.
3. In the row for the DGE-extension that you want to modify, click **Update**.
4. After making your desired changes to the DGE-extension properties, click **Update DGE Extension**.

Deleting a DGE Extension

1. Log in to the **Traverse** Web Application as `superuser`.
2. Navigate to Superuser > **DGE Mgmt**.
3. In the row for the DGE-extension that you want to modify, click **Delete**.
4. Click **Delete DGE Extension** if you are sure you want to permanently delete the extension, otherwise you can click **Cancel**.

> **Note:** Deleting the DGE-extension also deletes all devices and tests and corresponding historical data associated with it.
>
> **Note:** If you replace a DGE-extension with a DGE, you will lose the historical data that is stored at the upstream DGE. Please contact **Kaseya Support** *(https://helpdesk.kaseya.com/home)* if you want to do this type of conversion.

Monitoring DGE Extensions

Adding the 'Time Since Result From DGE Extension' Test

Once a DGE-extension has been configured and is publishing results, a **Time Since Result From DGE Extension** test must be provisioned on its upstream DGE to monitor if the DGE-extension is working. This is setup as a `test' of the upstream DGE itself.

1. The DGE itself must be added as a device to **Traverse** to be monitored by **Traverse**, preferably by another DGE if one exists.
2. Navigate to Administration > **Devices** and click on the **Tests** link of the upstream DGE.
3. Click on the **Create New Standard Tests** and select the last option **Create new test by selecting specific monitors**.
4. Check the box after JMX and **Add Tests**.
5. On the next page, select an existing JMX monitoring instance if one exists for port `7692`, else create a new instance specifying `7692` for the port number and leaving all other fields blank.
6. From the **Test Categories** box, deselect all and scroll down to select only **Traverse: [DGE] Time Since Result From DGE Extension** and click **Continue**.
7. On the next screen, provision all tests similar to **Time Since Result From DGE Extension** (name of DGE-extension) for each DGE-extension that this device is an upstream DGE.

Increasing Data Spool Time Period of a DGE-extension

When an upstream DGE is not reachable, the DGE-extension automatically spools the monitored data until a specified time. To change this value, edit `etc\datacollector.xml` on the system hosting the DGE Extension and increase the `timetoLive` value which is specified in milliseconds.

Consult with **Kaseya Support** *(https://helpdesk.kaseya.com/home)* before increasing this value.

Managing DGEs

DGE Locations and Management

You can manage DGEs and DGE-extensions through the **Traverse** Web Application by navigating to Superuser > **DGE Mgmt**. From there, you can see a list of all of your locations and DGEs, as well as license information. The number of devices and tests for all DGEs and DGE-extensions are shown and automatically added so you can see the total number of provisioned devices and tests.

DGE Locations and Management

Select an operation for a DGE location and management below or use the link to create a new location and dge.

Create New Location

Create New DGE

DGE Extension Management

Create New DGE Extension

DGE Locations

NAME	DGE COUNT	STREET	CITY	STATE	COMMENTS	MODIFY
Default Location	1					Update Delete

DGE Management

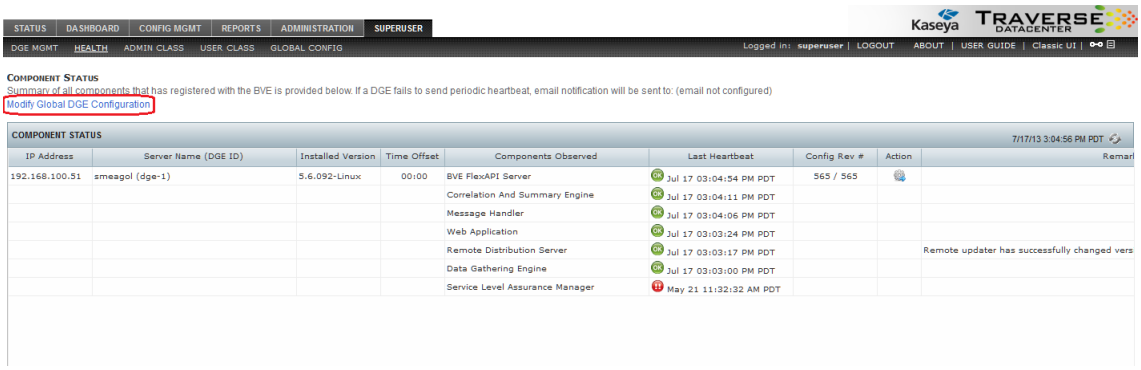
NAME	HOST	LOCATION	DEVICE COUNT	TEST COUNT	SOFT/HARD LIMIT	MODIFY
dge-1	127.0.0.1	Default Location	9	406	35000 / 50000	Update Delete
dge-ex1		dge-ex1:sample dge extension	1	2	15000 / 20000	Update Delete
Total for DGE			10	408		
Grand Total			10	408		

License Parameters

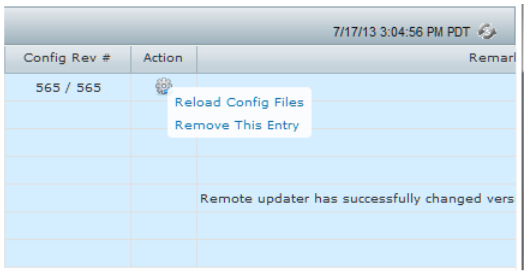
Number of DGEs	DGE Ext.	Ping Devices	Simple Devices	Standard Devices	Total Devices	Total Tests
16	16				unlimited	unlimited

Monitoring DGE Operation and Capacity

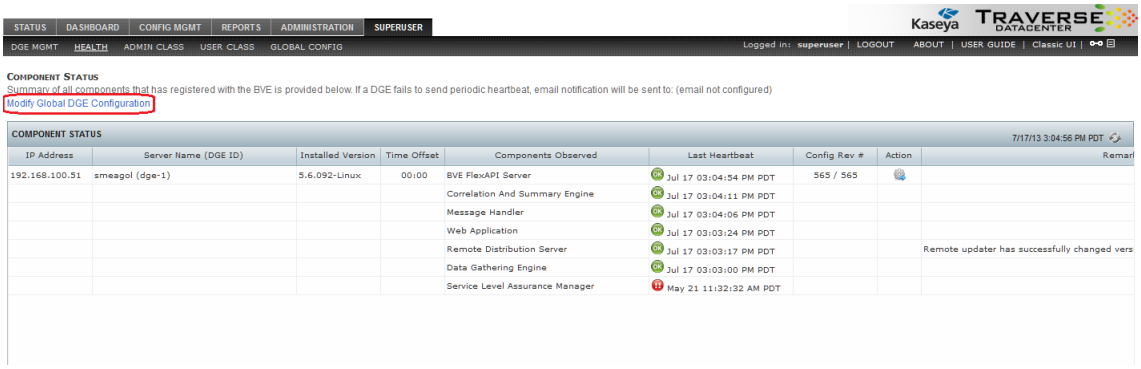
The components of **Traverse**, including DGEs and DGE extensions, can be easily monitored and their status checked from the Superuser > **Health** screen.



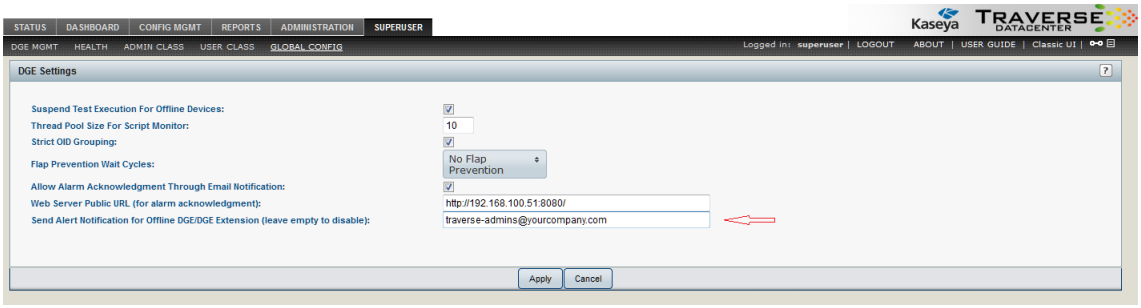
This page presents you with a list of the running components observed, their state, and the last date/time that the component provided a heartbeat. Additionally, the configuration revision of the local configuration files, and any remarks are presented. In the action column, you may choose to remove the server and all of its components from the health screen, or to reload the configuration files.



Removal of the entry clears all components for the server. Once a server has checked back in, the entry reappears, with the most current status on the health screen. From this screen, you may also setup notifications for DGE or DGE extensions that have stopped reporting in. First, choose the **Modify Global DGE Configuration** link, from the **Component Status** (or **Health**) page.



On this page, you can specify an email recipient to send alerts to in the event of a DGE or DGE extension going offline. This field is empty by default, thus sending no messages in the event of a downed DGE or DGE extension.



In addition to the user interface elements provided for monitoring the DGE, the DGE component itself keeps track of different types of monitors that are running, the number of objects processed and the number of items in various queues waiting to be processed.

Monitoring the Status of the DGE Using Telnet

You can telnet into the DGE component. Use port 7655, the default, or the port you have configured on the server.

```
% telnet my_dge 7655
Trying n.n.n.n...open
Connected to my_dge
Escape character is '^]'.
Traverse device monitor
password: *****
<<welcome>>
```

Once logged in, you can use the status command to view the health of each monitor, as well as the number of times they have performed a health check of configured elements.

```
controller> status
<<begin>>
Monitor[sql] - com.fidelia.emerald.monitor.SqlQueryMonitor
Number of passes: 0
Work Units processed: 0
Thread Status: alive
Monitor[radius] - com.fidelia.emerald.monitor.RadiusMonitor
Number of passes: 993
Work Units processed: 993
Thread Status: alive
Monitor[ldap] - com.fidelia.emerald.monitor.LdapMonitor
Number of passes: 0
Work Units processed: 0
Thread Status: alive
[additional status lines removed]
<<end>>
```

On a healthy DGE, **Thread Status** for all the monitors should indicate alive and the number of passes and number of work units processed should be increasing, provided there are one or more tests of that particular type configured (and not suspended) in the system.

The DGE status server also provides important information regarding capacity planning. The **Schedule Queue** section of the status command output indicates how many tests are waiting to be performed:

```
MonitorServer
Schedule Queue [Monitor[sql]] Size: 0
Schedule Queue [Monitor[ldap]] Size: 0
Schedule Queue [Monitor[radius]] Size: 0
Schedule Queue [Monitor[port]] Size: 0
Schedule Queue [Monitor[ntp]] Size: 0
Schedule Queue [Monitor[poet]] Size: 0
Schedule Queue [Monitor[ping]] Size: 0
Schedule Queue [Monitor[snmp]] Size: 2
Schedule Queue [Monitor[dns]] Size: 0
Schedule Queue [Monitor[external]] Size: 0
Result Queue Size: 0
Aggregation Writer Queue Size: 0
Result Writer Queue Size: 0
Event Writer Queue Size: 0
```

In the event of a network outage, the size of different queues may grow to a large number depending on the network topology and reachability of each device. Once the outage has been resolved, the queues should start to decrease. However, if under normal operating conditions the queue continues to grow, it would indicate that new tests are being added to the queue before existing tests can be performed, and your DGE capacity has been exceeded. At this point you need to one of the following:

- Add another DGE at the same location.
- Move some tests/devices to a different DGE, either at same location or a different location.

- Reduce the frequency of the tests or suspend some tests until capacity on the DGE can be increased.

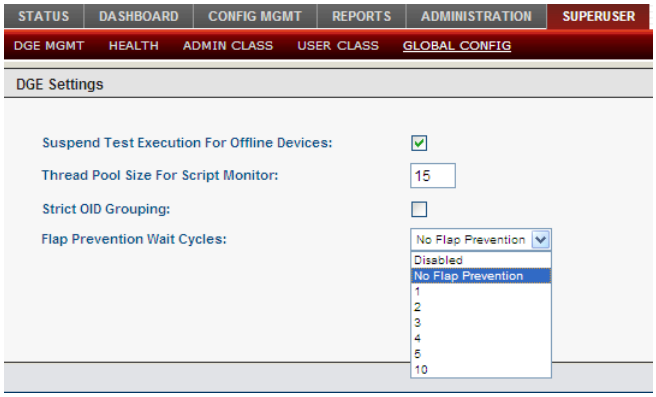
Once completed, you can use the `quit` command to log out of the DGE status server.

```
controller> quit
<<bye>>
Connection closed by foreign host.
```

DGE Global Configuration

There are a few global DGE settings that you can change by navigating to Superuser > Global Config > **Data Gathering Engine.**

- **Suspend Test Execution For Offline Devices**
- **Thread Pool Size For Script Monitor**
- **Select OID Grouping -** Ensures that all SNMP tests for the same device are scheduled together and therefore placed in the same batch. This can be useful for special cases, but this option should not be enabled unless a thorough analysis of your environment has been performed.

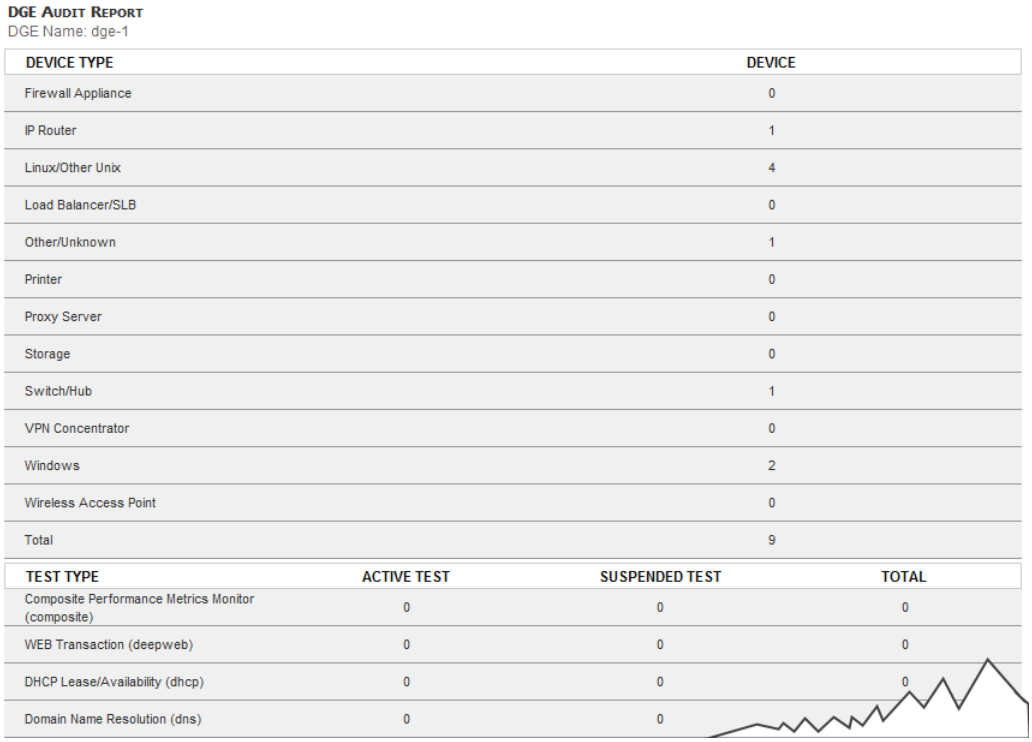


- **Flap Prevention** - Normally when a test or device going into warning or critical state, this state is shown immediately on the display. You can set the notification to be done immediately or after a few polling cycles. You can set the display state to be transient, instead of immediately displaying Warning or Critical to prevent flapping tests from displaying on the screen. This can be done on a global "per DGE" basis, or on a per device or per test basis. The global setting for each DGE is configured on this page, and you can override this for each device or test on the configuration page for each device or test also.

  A sample use case is to set the global default to be 2 polling cycles, and then for all the critical devices and tests, set the **Flap Prevention** value to 0 so that they show up on the screen right away. Note that this is only a display parameter, and does not impact reports and actions.

DGE Audit Report

You can get a report on the devices and tests provisioned on a DGE by navigating to Superuser > **DGE Mgmt** and clicking on the DGE name. This report shows the number of each type of device and the number of tests on the DGE.



DGE Audit Report
DGE Name: dge-1

DEVICE TYPE	DEVICE
Firewall Appliance	0
IP Router	1
Linux/Other Unix	4
Load Balancer/SLB	0
Other/Unknown	1
Printer	0
Proxy Server	0
Storage	0
Switch/Hub	1
VPN Concentrator	0
Windows	2
Wireless Access Point	0
Total	9

TEST TYPE	ACTIVE TEST	SUSPENDED TEST	TOTAL
Composite Performance Metrics Monitor (composite)	0	0	0
WEB Transaction (deepweb)	0	0	0
DHCP Lease/Availability (dhcp)	0	0	0
Domain Name Resolution (dns)	0	0	

Upgrading DGE Hardware

As the load on a DGE increases, it may be necessary to perform upgrades to the capacity of the hardware to increase the physical limits of the machine. If the upgrade involves addition of resources—memory, disk space, etc.—to the same machine, no special steps need to be performed. However, if the physical server is being upgraded for any reason, then the following steps need to be performed. Refer to the section on database backup/restoration for additional details.

Moving a DGE to a New Host

1. Install **Traverse** on the new host, making sure to use the same DGE name as the old one when asked during the install. Otherwise you will need to edit `<TRAVERSE_HOME>/dge.xml` and configure the DGE name there.
2. If the new host will have a different IP address, then you also need to log in to the Web Application as `superuser`, navigate to Superuser > **DGE Management**, and change the IP address of the relevant DGE.
3. Copy the following directories and files in their entirety from the old host to the new host:

```
database/
plugin/
etc/licenseKey.xml
etc/dge.xml
etc/emerald.xml
```

4. Restart the new DGE.

Chapter 5

Additional Notification Features

In This Chapter

Alphanumeric Paging

Note: This feature is only available in **Traverse** on premise.

To send notifications to a pager using a directly attached modem, select **Alphanumeric Pager** from the **Notify Using** list. Then, specify a **Message Recipient** in the format `PIN@PC`, where PIN is the recipient's Personal Identification Number (usually the pager number) and PC is a 'paging central' location defined by your **Traverse** administrator for the DGE that will generate the notification. See **Action Profiles** *(http://help.kaseya.com/webhelp/EN/TV/9020000/index.asp#17491.htm)* for detailed instructions.

For example, assume that an administrator has set up a paging central location called `sprint` that is used when **Traverse** sends pages to Sprint customers. A typical pager message recipient might be `7325551212@nextel`.

The notification content can be customized on a global basis by the **Traverse** administrator and described in.**Customizing the Notification Content** *(page 49)*.

Configuring Alphanumeric Paging

Traverse can send alphanumeric messages to a TAP/IXO pager using a modem attached to the DGE. Note that each DGE can have one or more locally attached modems, which ensures maximum redundancy and fault tolerance in a distributed environment.

Configuring Traverse for Alphanumeric Paging

1. Edit the `<TRAVERSE_HOME>/etc/emerald.xml` configuration file on the DGEs and add modem configuration information as described in **Modem Configuration** *(page 44)*, and paging central information for the paging service provider as described in **Paging Central Software Configuration** *(page 45)*.
2. Edit the `<TRAVERSE_HOME>/etc/emerald.xml` configuration file on the Web Application and copy the paging central information for all the DGEs into this file. This Paging Central list is displayed in the **Action Profiles** drop down.
3. Create action profiles that use alphanumeric paging as described in **Administrator Configured Action Profiles and Thresholds** *(http://help.kaseya.com/webhelp/EN/TV/9020000/index.asp#17504.htm)*, and assign them to tests that are run by this DGE.
4. Restart the Web Application and the DGE components.

Modem Configuration

You can have multiple modems attached to a DGE. For each modem that's attached to the DGE, add a modem-config section to the DGE's `<TRAVERSE_HOME>/etc/emerald.xml` file. If multiple modems are configured, they are used in the order specified by their device priority parameters. Tthe lower the number, the higher the priority. For each modem, set the following parameters.

Parameter	Description
sender id	The phone number used to identify this modem when sending a page. You can set it to any phone number representing this DGE.
device priority	This modem's priority with respect to other modems attached to the DGE. The lower the value of this parameter, the higher the modem's priority. When sending a page, Traverse uses the highest-priority modem that is available.
port	The port through which this modem communicates. UNIX: Enter a port in the format `/dev/ttySn` where n is 0,1,2.

	Windows: Use the format COMn where n is the number of the COM port.
speed	The modem's transmission speed, expressed in bits per second.
parity	The type of parity checking, if any, used by this modem. Possible values are even, odd, and none.
databits	The number of data bits transmitted in each series. Possible values are 7 and 8.
stopbits	The number of bits used to indicate the end of a byte. Possible values are 1, 1.5, and 2.

Example of Modem Configuration in emerald.xml

```
<modem-config>
 <sender id="3035557777"/>
 <device priority="10">
  <port>/dev/ttyS0</port> <!-- /dev/ttyS or COMn -->
  <speed>9600</speed> <!-- bps -->
  <parity>none</parity> <!-- none, odd, even -->
  <databits>8</databits> <!-- 8, 7 -->
  <stopbits>1</stopbits> <!-- 1, 1.5, 2 -->
 </device>
</modem-config>
```

Paging Central Software Configuration

Every paging service provider has its own central number and modem pool configuration. For each paging service provider that will be used, add a paging-central child element to the alpha-pager element of the DGE's `<TRAVERSE_HOME>/etc/emerald.xml` file. For each service provider, set the following parameters:

Paging Configuration Parameters

Parameter	Description
name	A name that uniquely identifies this service provider to the DGE.
number	The number the DGE must dial to reach Paging Central, including any prefixes. You can find many Paging Central phone numbers at **http://www.notepager.net/tap-phone-numbers.htm** *(http://www.notepager.net/tap-phone-numbers.htm)* or a similar site.
speed	The highest speed supported by the service provider. Possible values include the following: • 0 (110bps) • 2 (300bps) • 4 (1200bps) • 5 (2400bps) • 6 (4800bps) • 7 (9600bps) The default value is 5.
parity	The type of parity checking, if any, supported by the service provider. Possible values include the following: • 0 (none) • 1 (odd) • 2 (even) • 3 (mark) • 4 (space) The default value is 2.
databits	The number of data bits supported by the service provider. Possible values are 2 (7

	bits) and 3 (8 bits). The default value is 2.
stopbits	The number of end-of-byte bits supported by the service provider. Possible values are 1, 1.5, and 2. The default value is 1.
flowcontrol	The type of handshaking supported by the service provider to prevent data loss during transmission. Possible values include the following: • 0 (none) • 1 (XONXOFF) • 2 (CTSRTS) • 3 (DSRDTR) The default value is 2.

The `alpha-pager` parent element also includes a sender id, which identifies the modem that is used to communicate with the specified paging central locations, as well as one or more device priority child elements that specify what port is used.

Note that it is typical to have several `<paging-central>` definitions since your staff might have pagers (cell phones) from different vendors, and each vendor has their own phone number for paging. While creating action profiles, the vendor is specified using the `pager-pin@pager-central-name` syntax.

If the modem is not available or busy, pages are queued on the DGE. Undeliverable pages older than 1 hour are ignored. Tthese parameters can be controlled via the configuration in `emerald.xml` also.

Example Paging Configuration in emerald.xml

```
<alpha-pager>
 <sender id="3035557777"/>
 <device priority="10" port="/dev/ttyS0" />
 <device priority="20" port="/dev/ttyS2" />
 <paging-central name="attws"> <!-- name should be unique -->
   <number>9998887777</number> <!-- number to dial, including prefix -->
   <speed>5</speed> <!-- 0=110bps, 2=300bps, 4=1200bps 5=2400bps, 6=4800bps,
7=9600bps -->
   <parity>2</parity> <!-- 0=none, 1=odd, 2=even, 3=mark, 4=space-->
   <databits>2</databits> <!-- 2=7bits, 3=8bits -->
   <stopbits>1</stopbits>
   <flowcontrol>1</flowcontrol> <!-- 0=none, 1=xonxoff, 2=ctsrts 2=ctsdtr, 3=dsrdtr
-->
 </paging-central>
 <paging-central name="nextel"> <!-- name should be unique -->
   <number>3035551212</number>
   <speed>5</speed>
   <parity>0</parity>
   <databits>3</databits>
   <stopbits>1</stopbits>
   <flowcontrol>0</flowcontrol>
 </paging-central>
</alpha-pager>
```

SMS or Cell Phone Messaging

> **Note:** This feature is only available in **Traverse** on premise.

You can send an SMS to a cell phone using supported SMS modems such as the **MultiTech iSMS Gateway** *(http://www.multitech.com/manuals/s000461f.pdf)*. Integration details are described below.

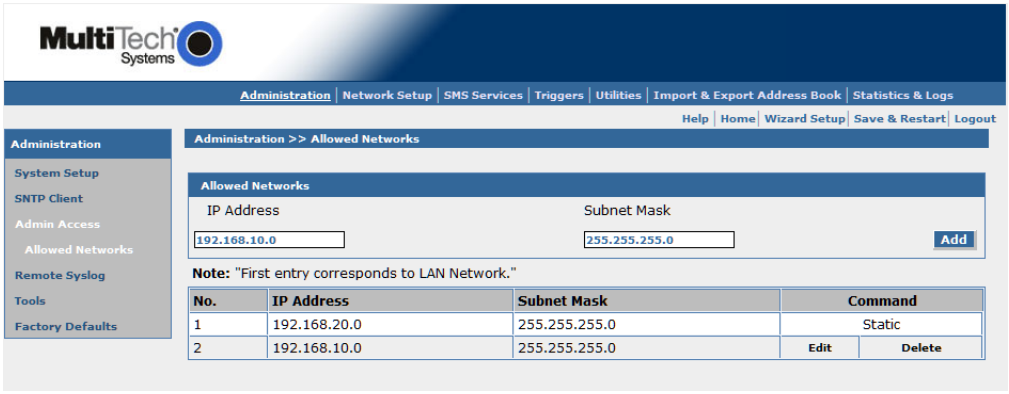
You can integrate with the MultiTech iSMS gateway modems (SF100, SF400) to send SMS messages

to cellular phones for notifications. These SMS modems have an ethernet network interface, so they can be shared by multiple DGEs and are easily serviceable. The same procedure described here can be used to integrate with other SMS modems.

After completing the initial setup of the SMS modem and testing if it can send messages, you need to do the following additional steps to integrate with **Traverse**.

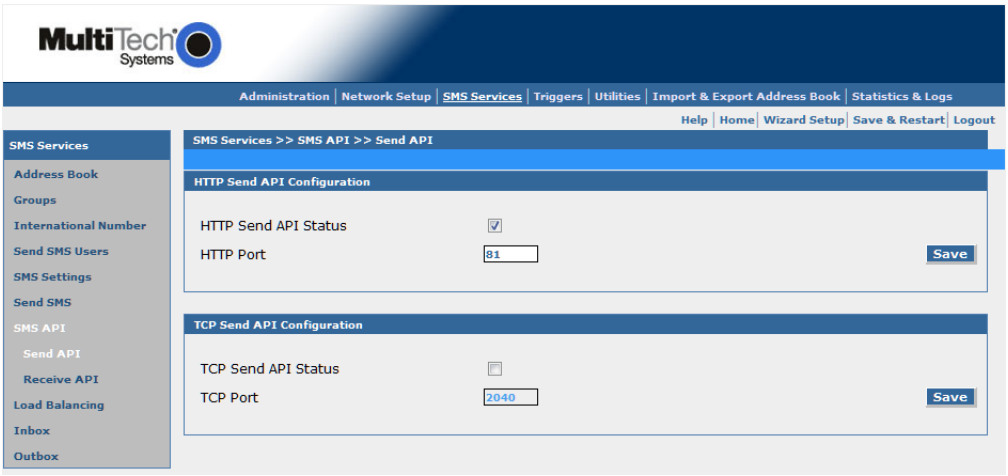
Allow the DGE Access

- On the Multitech modem specify the network you wish to send alerts from via the API.
- Login to the **Multitech Administrative Page**, and select **Administration** from the top menu.
- Select **Admin Access, Allowed Networks** from the left hand menu.
- In the **Allowed Networks** box, enter the network, and subnet mask for any DGE's you wish to be able to alert from, and press the **Add** button.



Enable the HTTP Send API

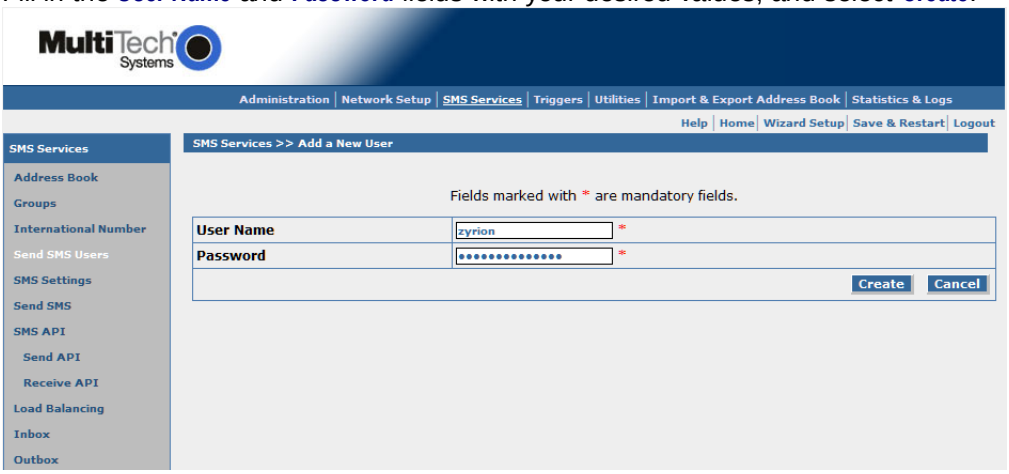
- Next, choose **SMS Services** from the top menu.
- On the left hand menu, select **Send API**, underneath the **SMS API** entry.
- Enable the **HTTP Send API Status** selection, and choose an appropriate port (81 is the default).
- Select **Save**.



Create a user for the Plugin

- Select **SMS Services** from the top menu. You should already see **SMS Services** displayed for you.
- From the left menu, choose **Send SMS Users**.

- Press **Add**.
- Fill in the **User Name** and **Password** fields with your desired values, and select **Create**.



Install and Configure the Traverse Action Plugin

- On the DGEs that will be sending SMS alerts, unzip the `multitech-isms` package into the `TRAVERSE_HOME/plugin/actions` directory.
- Edit the file `multitech-isms.pl` and replace the following values with what you have configured for your environment.

iSMS Plugin Action

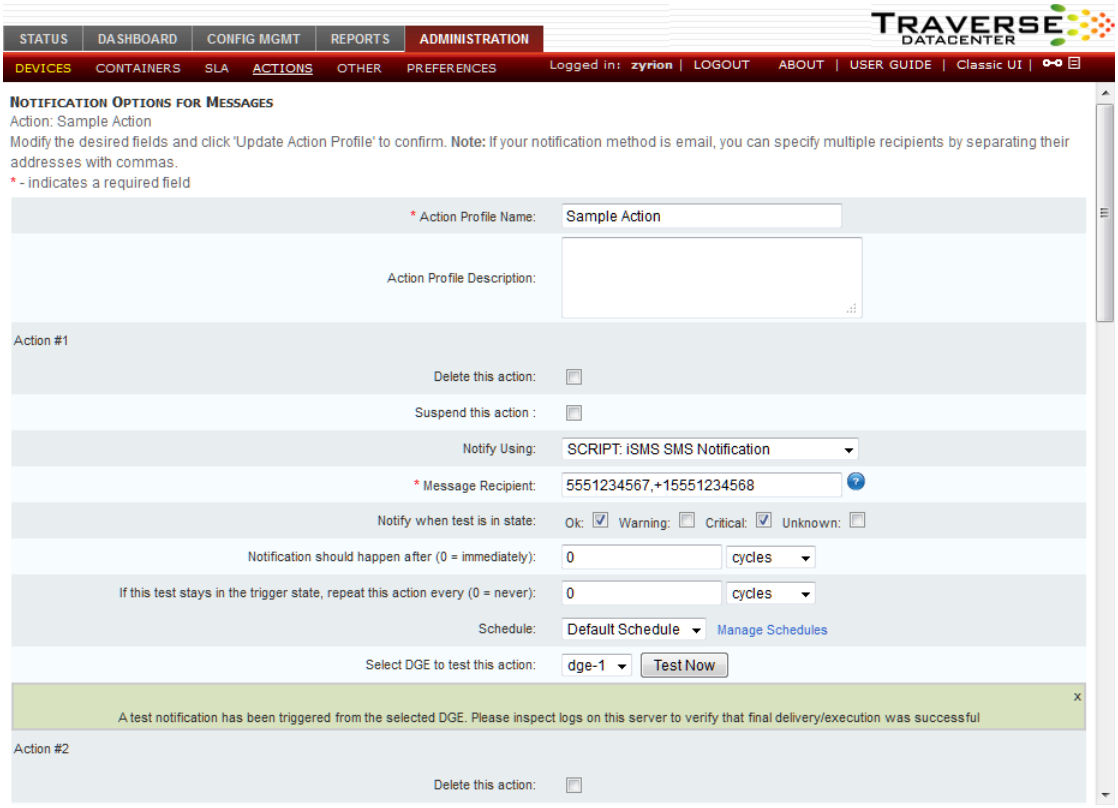
```
##############################################
#### USER CONFIGURABLE OPTIONS ARE HERE  ####
##############################################

# These are the defaults for the device, and should work
# if the user makes no changes.
my $iSMS     = "192.168.2.1";
my $iSMSPort = "81";
my $iSMSUser = "admin";
my $iSMSPass = "admin";

##############################################
####    END USER CONFIGURABLE OPTIONS    ####
##############################################
```

Restart your DGE and Web Application to load the plugin.

In the **Traverse** Web Application, you should now have the ability to use `SCRIPT: iSMS SMS Notification` as a **Notify Using** type in your action profiles.



Note: Multiple SMS recipients are allowed, by separating the phone #'s with a comma. This plugin does not use the address book, or groups defined in the iSMS admin pages, but support can be added relatively easily. Phone numbers can be specified in any format, including north american with () and spaces, such as (555) 123-4567.

Customizing the Notification Content

The notification content for the built-in notifications can be customized by editing the following files in the `<TRAVERSE_HOME>/etc/actions/` directory:

- `regular-email.xml`
- `compact-email.xml`
- `tap-pager.xml`

There can be two sections in each file, one for the test threshold violations (type="test") and one for the traps and log messages (type="message"). All the variables that are used in the plugin framework (see the **Traverse Developer Guide & API Reference** *(http://help.kaseya.com/WebHelp/EN/tv/9020000/DEV/index.asp#home.htm)*) are available for these notification XML files as well.

All multi-line text in the `<body>` parameter is combined into a single line. Any `\r\n` or `\n` strings are converted into newline characters unless they are prefixed by a `^` character. For example, the configuration

```
a\r\n
b
c ^ d \n
```

is converted to

```
a
bc d
```

Chapter 6

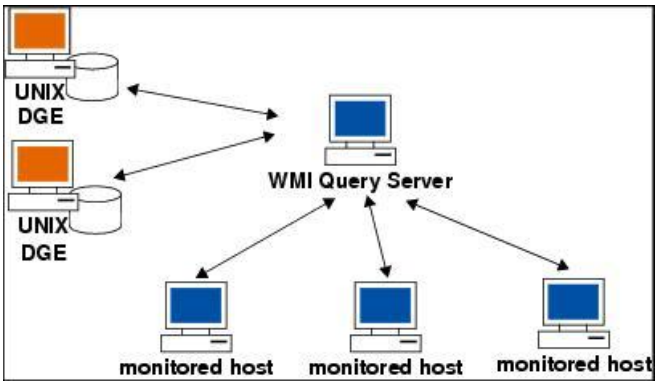
Configuring WMI in Unix

This feature is only available in Traverse on premise.

In This Chapter

WMI Monitoring from a UNIX DGE

To perform WMI monitoring from a UNIX DGE or DGE extension you must install and configure a **Traverse** WMI Query Server as a "proxy" on a Windows system that can access the Windows hosts to be monitored. Note that there is a corresponding WMI Event Listener program (`nvwmiel`) to monitor Windows events using WMI. See **The Traverse WMI Event Listener** *(http://help.kaseya.com/webhelp/EN/TV/9020000/index.asp#17211.htm)*.



Relationship Between DGEs, WMI Query Server, and Monitored Hosts

Traverse WMI Query Server Installation for Traverse on UNIX

Note: You only need to install the WMI Query Server if your Traverse installation is on a UNIX server. *The WMI Query Server is automatically installed on Windows DGEs and DGE extensions.*

The WMI Query Server (`nvwmiqd`) should be installed on a Windows machine which has access to the Windows hosts being monitored using NetBIOS.

WMI Query Server System Requirements

Install the **Traverse** WMI Query Server on a system that meets or exceeds the following requirements:

- Pentium III processor or greater, 512MB RAM, 25MB free disk space
- Windows 2000/2003/XP/Vista (English only)

Installing the Traverse WMI Query Server

1. Test if the Windows machine you want to install the WMI Query Server on has access to the Windows hosts being monitored by typing the following at a Windows command prompt:

```
NET VIEW \\remote_host
```

2. Download the WMI Query Server (`wmitools-x.y.z-windows.exe`) from the **Traverse** CD-ROM or the **Kaseya Support** *(https://helpdesk.kaseya.com/home)* site and save the file to a temporary directory. For example: `C:\temp`.
3. Double-click `wmitools-x.y.z-windows.exe`.
4. Read the **Introduction**, and then click **Next** to continue.
5. Optionally, in the **Choose Install Folder** window, specify the folder in which you want to install the WMI Query Server. Click **Next** to continue.

6. In the **Pre-Installation Summary** window, review the configuration options. If they are correct, click **Install** to continue.
7. After the installation completes, click **Done** to close the installer.

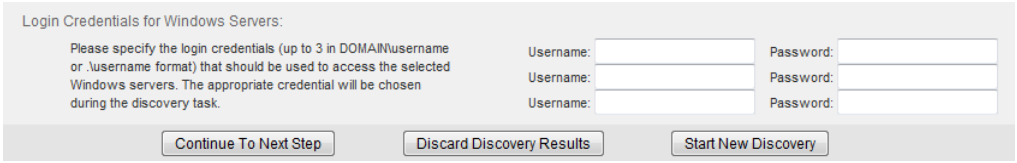
Configuring the WMI Query Daemon to Run Under a Different Account

If you do not specify login credentials when you add a test to a device, the WMI Query Server by default uses the username and password of its local system account to access the monitored Windows host. However, this account typically does not have the necessary rights to access remote Windows servers. If you want the WMI Query Server to use a specific account as the default, do the following steps.

1. Navigate to Start > **Run**.
2. Execute `services.msc`.
3. Double-click the **Traverse** WMI Query Daemon service in the list of services.
4. Click the **Log On** tab.
5. Select **This account**, provide the username and password for the credentials you want the service to use, and then click **OK**.
6. Restart the **Traverse** WMI Query Daemon service.

Access Requirements

Each Windows host that you want to monitor through WMI must have a user account that the WMI Query Server can access (with administrative permissions to access various system tables). You can specify these credentials after performing a discovery in the **Traverse** Web Application (Administration > Other > Device Discovery & Import > **New Network Discovery Session**).



Enter credentials for up to three Windows domains.

Examples:

```
DOMAIN1\username password
DOMAIN2\username password
.\username password
```

Note: Enter an administrator username in `.\username` format to access Windows systems that do not belong to a Windows domain.

- If the Windows host to be monitored is part of a domain, you will need the username and the corresponding password for a user who is part of the Domain Administrator group. The WMI Query Server will use this user's credentials to connect to the Windows hosts being monitored for retrieving the WMI performance information.
- If the hosts are configured in one or more workgroups, and not part of a domain, then each host, including the host where the WMI Query Server is being installed, will need to have the same password for the administrator user, or have another such common user which is part of the Administrators group.

DGE Configuration for Proxy WMI Server

If you have any UNIX DGEs which need to use the WMI Query Server on a Windows machine as a

proxy, edit the following parameters in `<TRAVERSE_HOME>/etc/dge.xml`:

```
<wmiQueryServer>
<host name="my_host_1" address="1.1.1.1" port="7667" username="wmiuser"
password="fixme" />
</wmiQueryServer>
```

Restart the DGE so that the changes can take effect.

dge.xml Parameters

The parameters in the `dge.xml` file are as follows.

Parameter	Description
`host name`	A unique, descriptive name for the WMI Query Server host that this DGE uses for WMI monitoring (e.g., Denver_WMI_QueryHost).
`address`	The IP address of the WMI Query Server host, in dotted quad notation. If the DGE is running on Windows, this will be set to 127.0.0.1
`port`	The TCP port on the WMI Query Server to which the DGE connects. This must match the `port` parameter in the `nvwmiqd.ini` file on the WMI Query Server.
`username`	The username that the DGE uses to log in to the WMI Query Server. This must match the `username` parameter in the `nvwmiqd.ini` file on the WMI Query Server.
`password`	The password that the DGE uses to log in to the WMI Query Server. This must match the `password` parameter in the `nvwmiqd.ini` file on the WMI Query Server.

You can have up to 4 DGEs using a single WMI Query Server as a proxy.

Chapter 7

Traverse Configuration Files

In This Chapter

Overview

The **Traverse** system uses several configuration files to obtain information about different components and system parameters. Before starting the application, you need to make sure that the default values match your local network and server configurations in the files described below.

These configurations can be applied to any DGE or DGE extension you have access to. Their scope applies only to the devices being monitored on their network.

Application Installation Path (UNIX Only)

Configuration File

```
<TRAVERSE_HOME>/etc/emerald.env
```

Restart These Components After Changing the Configuration File

- Provisioning Database
- Web Application
- Monitor

Description

This file contains environment variables that specify the location of different supporting software required to operate **Traverse**. `INSTALL_DIR` should be set to the installation directory `<TRAVERSE_HOME>`. Do not modify other variables unless instructed to do so by **Kaseya Support** *(https://helpdesk.kaseya.com/home)*.

BVE Config Database Host/Location

Configuration File

```
<TRAVERSE_HOME>/etc/emerald.xml
```

Restart These Components After Changing the Configuration File

- Web Application
- Monitor

Description

Monitors that are part of the DGE component and web interface use this file to identify the Provisioning Database. If the DGE or Web Application component is operating on the same server as the Provisioning Database, you do not need to change this file. Otherwise, edit the following line:

```
<provisioning
name="provisioning"
host="localhost"
[...]
```

Change `localhost` to the fully qualified domain name (FQDN) or IP address of the server where you are planning on operating the Provisioning Database. Do not change the user and password parameters.

Logging Configuration

Configuration File

`<TRAVERSE_HOME>/etc/log4j.conf`

Restart These Components After Changing the Configuration File

- Provisioning Database
- Web Application
- Monitor

Description

Different components of **Traverse** provide useful diagnostic and informative log messages. You can specify the amount of logged information by changing `LOGLEVEL` to one of the following parameters in the following table.

Log Message Detail Levels

LOGLEVEL	Level of Detail
`INFO`	Informational messages that highlight the progress of the application at a coarse-grained level.
`WARN`	Designates potentially harmful situations.
`ERROR`	Designates error events that might still allow the application to continue running.
`FATAL`	Designates very severe error events that will presumably lead the application to abort.
`DEBUG`	Additional detailed information that is useful for debugging an application. Do not enable debug messages unless asked to do so by **Kaseya Support** *(https://helpdesk.kaseya.com/home)*.

By default, **Traverse** only logs messages into log files stored in the directory specified by the `$LOGDIR` variable. If you want to send logs to a UNIX syslog host at a central location or on same hosts, uncomment the following section:

```
#log4j.appender.SYSLOG = org.apache.log4j.net.SyslogAppender
#log4j.appender.SYSLOG.SyslogHost = localhost
#log4j.appender.SYSLOG.facility =
org.apache.log4j.net.SyslogAppender.LOG_LOCAL7
```

Then change `localhost` to the FQDN or IP address of the host to which you want to send the log messages. If you want the messages sent as a facility other than local7, change `LOG_LOCAL7` to `LOG_FACILITY` where `FACILITY` is one of the facilities listed in the man page (man5) of `syslogd.conf`. Make sure to enter the facility name in upper case.

Test Definitions and Defaults

Configuration File

`<TRAVERSE_HOME>/etc/TestTypes.xml`

Restart These Components After Changing the Configuration File

- Provisioning Database
- Web Application

Description

This file contains information on default values for thresholds and display properties of various tests. When **Traverse** provisions new tests, or displays existing test results, information in this file

determines how to group similar tests together and the units to use to display test results. The file is in XML format and the formatting must be maintained while making any changes to the file.

The provisioning server and Web Application use this information when you do not specify thresholds in the Web Application. When you specify default thresholds for any department, **Traverse** stops using this file to populate default thresholds when you create tests for that particular department.

See the **Traverse Developer Guide & API Reference** *(http://help.kaseya.com/WebHelp/EN/tv/9020000/DEV/index.asp#home.htm)* for more information about this file.

External Help

External help provides **Traverse** operators the ability to write support documentation specific to a department, device or test and tie it directly to that same object via a **Help** link in the web user interface. This way, less experienced system administrators can be provided with a first line of troubleshooting in the absence of live support. You can also enable actions (e.g., server restart) via the **Help** links. This is a powerful option, as any number of files can be configured to work in this fashion, enabling a large number of background processes via the Web Application.

The default `<TRAVERSE_HOME>/utils/externalTestHelp.pl` perl script scans through the entire `<TRAVERSE_HOME>/plugin/help` directory tree for help text specific to a department, device or test. This script expects one argument in the following form:

```
<department_name> | <device_name> | <device_addr> | <test_type> | <test_subtype> | <test_name>
```

where `device_addr` can be FQDN or IP address. This has to match what was used for device creation. The field `test_name` should match the descriptive name that was displayed during test creation (or in test details page).

> **Note:** The perl script converts everything (for example, `acct_name` and `device_name`) to lowercase to avoid any case related problems when searching for the file. One or more consecutive space characters in device names and test names are converted to an underscore (_) character. Therefore, the directories and subdirectories must be named in lowercase, spaces substituted with underscore, and special characters formatted the same as the department and device names.

The script searches `<TRAVERSE_HOME>/plugin/help` according to following algorithm:

1. Search for directory `acct_name ELSE use _default_user`
2. If found, cd into it.
3. Search for subdirectory `device_name ELSE device_addr ELSE _default_device`
4. If found, cd into this sub-directory.
5. Search for the files in the current directory in the following order:

```
<test_type>_<test_subtype>_<test_name>.{html,txt} ELSE
<test_type>_<test_subtype>.{html,txt} ELSE
<test_type>.{html,txt} ELSE
default.{html,txt}
```

6. Display the entire file on `stdout` (if text, then put HTML tags around the text).
7. If not found, display `NO FILE FOUND` on `stdout` in HTML format. The script prints out errors on `stdout`. The location of the script is specified in `web.xml` and it can basically be any script or program. It is up to the target script to take the arguments and send back help text in the required format.

For example, to create a help file for `device mail_server` and a more specific one for the `disk_space`, in department `local_department:`

```
cd <TRAVERSE_HOME>/plugin/help
mkdir -p local_department/mail_server
mkdir -p local_department/_default_device
cd local_department/
vi _default_device/default_html
vi mail_server/snmp_disk.txt
vi mail_server/default.html
```

It is possible to use your own script, that, for example, connects to a database and retrieves escalation information based on specified criteria.

Web Application External Help

Configuration File

`<TRAVERSE_HOME>/webapp/WEB-INF/web.xml`

Restart These Components After Changing the Configuration File

- Web Application

Description

Traverse allows you to add information to the Help link that is associated with each test item. When you click the **Help** link, you can display;

- escalation information
- procedures
- any information related to individual tests
- any information on a global basis related to test type, device, or department context

Traverse includes a default script (`<TRAVERSE_HOME>/utils/externalTestHelp.pl`) which scans for this information within in a directory hierarchy.

You can also obtain this information by executing an external script. Locate the following section in the `web.xml` file and modify it to specify the location of the script:

```
<param-name>help.script.path</param-name>
```

See **External Help** *(page 58)* for information about the algorithm used to find test-specific information.

Web Application URL Embedded Authentication

Configuration File

`<TRAVERSE_HOME>/webapp/WEB-INF/web.xml`

Restart These Components After Changing the Configuration File

- Web Application

Description

Traverse makes it easy to integrate the Web Application into an existing web portal or single-login system. Using the external authentication mechanism, you can bypass the initial authentication web page and go directly into the device summary page. This is accomplished by encoding user department and login information in an md5 hash, using the shared key and passing into the authentication engine of the Web Application component. The

`<param-name>externalLoginKey</param-name>` section is used to configure a shared key for external URL based authentication. See the section on Authentication in the **Traverse Developer Guide & API Reference** *(http://help.kaseya.com/WebHelp/EN/tv/9020000/DEV/index.asp#home.htm)* for further details on setting this up.

> **Note:** It is highly recommended that you edit the shared key defined in `<param-value>payday</param-value>` and change the default to something different and secure. A sample CGI script (`<TRAVERSE_HOME>/utils/externalWebLogin.cgi`) illustrating how to use the external authentication mechanism is provided with **Traverse**.

Also see the **Traverse Developer Guide & API Reference** *(http://help.kaseya.com/WebHelp/EN/tv/9020000/DEV/index.asp#home.htm)* for using external authentication using Windows Active Directory, LDAP, Radius, etc.

DGE Identity

Configuration File

`<TRAVERSE_HOME>/etc/dge.xml`

Restart These Components After Changing the Configuration File

- Monitor

Description

The following entry sets the name a DGE identifies itself with against the Provisioning Database:

```
<dge name="my_dge" user="emerald" password="null"/>
```

The name `my_dge` should be changed to the name of the DGE that you have (or are going to) set up. The name does not need to be an FQDN, only something meaningful. However, you will need to use the same name when creating DGE information in the Provisioning Database using the superuser interface. For example, if you plan to have a DGE with the name `dge01.central` with an FQDN of `dge01.central.mycompany.com`, then `my_dge` should be replaced with `dge01.central`, and you must use the same DGE name when you create the DGE using the superuser interface. (For more information on creating DGEs, see **DGE Management** *(page 29)*).

DGE Controller Port/Password

Configuration File

`<TRAVERSE_HOME>/etc/dge.xml`

Restart This Component After Changing the Configuration File

- Monitor

Description

Each DGE process listens on a TCP/IP port for incoming connection requests and provides status on each of the monitors it supports. By default this port is set to 7655, but this can be configured by editing the following section:

```
<controller port="7655" password="fixme"/>
```

If you change the port from 7655 to something different, make sure that no other application running on the machine is going to bind to that port. You should also change the password `fixme` to a different and more secure password. You will use this password to log in to the status server.

EDF Server Port/Password

Configuration File

```
<TRAVERSE_HOME>/etc/dge.xml
```

Restart These Components After Changing the Configuration File

- Monitor
- External Data Feed

Description

Each DGE process listens on a TCP/IP port for incoming connection requests and allows integration with external tools utilizing the External Data Feed API. By default this port is set to 7657, but this can be configured by editing the following section:

```
<edfMonitor>
<port>7657</port>
<connections>1</connections>
<timeout>120</timeout>
<userName>edfuser</userName>
<password>fixme</password>
</edfMonitor>
```

If you change the port from 7657 to something different, make sure that no other application running on the machine is going to use that port. You should also change the password `fixme` to a different and more secure password. You will use this password along with the specified username to log in to the EDF server. The connections parameter configures the number of concurrent connections to the EDF server that should be allowed. If you expect to run a lot of external monitors that need to insert results into **Traverse**, this number should be set to a suitably large number.

Email servers

Configuration File

```
<TRAVERSE_HOME>/etc/emerald.xml
```

Restart These Components After Changing the Configuration File

- DGE
- Report Server

Description

The DGE and Report Server components need to know which email servers they should use to send notifications or reports via email.

```
Edit the following section in <TRAVERSE_HOME>/etc/emerald.xml:
<email-servers>
<sender address="traverse@your.domain" name="Traverse Alerter"/>
<host name="mail_server1" priority="10"/>
<host name="mail_server2" port="589" priority="30">
</email-servers>
```

Change `mail_server1 / mail_server2` to the FQDN of your local email server or the email server that you use for sending outgoing email. If you have more than one email server, you can add additional servers with a different priority value (the lowest priority server is preferred).

Create an email alias for the **Traverse** administrator, and set the sender address to this email alias. All alerts from **Traverse** will be sent from this sender address.

> **Note:** There is a separate email address setting in `emerald.env` used for getting administrative alerts from **Traverse** such as DGE process failure, backups, etc.

You should make sure that the email servers are configured properly to allow **Traverse** to relay email to any email address. (Please refer to your email server's administration guide for instructions on how to accomplish this). See **Actions and Notifications** *(http://help.kaseya.com/webhelp/EN/TV/9020000/index.asp#17489.htm)* for more details.

Authenticated SMTP Over Plain-Text

You can optionally specify a username and password for authenticated SMTP:

```
<host name="mail_server2" port="25" username="abc" password="xyz" priority="20"/>
```

Encrypted SMTP using TLS

You can add the following parameter so that **Traverse** uses encrypted TLS connections for sending email:

```
starttls="true"
```

If the SMTP server supports TLS, then during the initial SMTP handshake, **Traverse** BVE/DGE will switch to encrypted TLS connection for sending email.

Encrypted SMTP using SSL

As an alternative to TLS, you can also enable SSL encryption by specifying an SSL port in the mail server section:

```
sslport="nnn"
```

e.g. for Gmail, use:

```
<host name="smtp.gmail.com" priority="10" sslport="465" username="abc"
password="xyz">
```

If both `STARTTLS` and `SSLPORT` are specified for the mail server, then the `SSLPORT` entry is ignored.

Web Server TCP/IP Port

Configuration File

```
<TRAVERSE_HOME>/apps/tomcat/conf/server.xml
```

Restart These Components After Changing the Configuration File

- Web Application

Description

This is the configuration file for Jakarta Tomcat application server. By default, the **Traverse** Web Application will run on TCP port 80. If you already have another web server or another application using that port, you will need to configure the Web Application to run on an alternate port.

Configuring the Web Application Port

1. Edit `<TRAVERSE_HOME>/apps/tomcat/conf/server.xml` using a text editor and locate the following section:

```
<Connector
className="org.apache.coyote.tomcat4.CoyoteConnector" port="80" minProcessors="20"
maxProcessors="80"
Change port="80" to a new unused port. For example, port 8080.
```

2. Edit `<TRAVERSE_HOME>/webapp/WEB-INF/web.xml` and locate the following section:

```
<init-param>
<param-name>report.server.port</param-name>
<param-value>80</param-value>
```

3. Change `port="80"` to the same port number used in Step 1.
4. Save the file and restart the Web Application if already running.
5. Wait 15-20 seconds for the Web Application to initialize and use your web browser to connect to `http://your_traversetraverse_host:8080/` and you should see the **Traverse** login page.

Web Server Inactivity Timer

Pages under most menu options, such as **Administration**, timeout after a certain period of inactivity. Pages under the **Status** and **Dashboard** menu options do **not** timeout.

Configuration File

```
<TRAVERSE_HOME>/webapp/WEB-INF/web.xml (UNIX)
<TRAVERSE_HOME>\ apps\tomcat\conf\web.xml (Windows)
```

If the above Windows directory and file do not exist, the configuration file is:

```
C:\Program Files (x86)\Traverse\Tomcat\conf\web.xml
```

Restart This Component After Changing the Configuration File

- Web Application

Description

In order to change the web inactivity timer, edit the following section in the above configuration file:

```
<session-config>
<session-timeout>60</session-timeout>
</session-config>
```

The timeout is specified in minutes. A value of -2 will disable the timeout completely. Once updated, you will need to restart the Web Application.

Customizing Device Tag Labels

Configuration File

```
<TRAVERSE_HOME>/etc/emerald.xml
```

Restart This Component After Changing the Configuration File

- Web Application

Description

Traverse provides five customizable device tags, which you can define to meet your needs. For example, you can store information about where a device is located (city, state, building, room, rack) or what corporate group it belongs to (payroll, helpdesk, etc.) By default, these attributes are displayed with the labels Custom Attribute 1, Custom Attribute 2, etc. You can change these labels to more meaningful names by editing the following section:

```
<device-tags>
<tag index="1" description="Custom Attribute 1"/>
<tag index="2" description="Custom Attribute 2"/>
<tag index="3" description="Custom Attribute 3"/>
<tag index="4" description="Custom Attribute 4"/>
<tag index="5" description="Custom Attribute 5"/>
</device-tags>
```

Replace the description parameters with the labels that you want to see in the Web Application. For example:

```
<device-tags>
<tag index="1" description="City"/>
<tag index="2" description="State"/>
<tag index="3" description="Building"/>
<tag index="4" description="Room"/>
<tag index="5" description="Rack"/>
</device-tags>
```

Note: These definitions do not affect the way custom attributes are stored or used. They affect the display *labels* only for the tags.

Note: Upon upgrade of the **Traverse** software, the changes to the device tag labels must be reinstated since they are currently not preserved automatically across upgrades.

Secure Remote Access Gateway

Configuration Files

```
etc/emerald.xml on Web Application
etc/emerald.properties on DGE
```

Description

The following section in `etc/emerald.xml` on the Web Application allows setting up a secure tunnel from the Web Application to a remote DGE or DGE extension and connect to a remote router or server using telnet, ssh, VNC or rdesktop.

```
<remote-access>
<enabled>true</enabled>
<port>7654</port>
<connection-pool>
<size>20</size> <!-- # of concurrent sessions -->
<start>11701</start> <!-- ports 11701 - 11711 -->
</connection-pool>
<idle-timeout>900</idle-timeout> <!-- 30 minutes -->
<session-timeout>21600</session-timeout> <!-- 6 hours -->
<jms-broadcast-topic>traverse_sshbroadcast</jms-broadcast-topic>
</remote-access>
```

On the DGE, the remote access section is in the `etc/emerald.properties` file.

```
## remote access
traverse.tools.sshClient=/path/to/ssh/client
traverse.tools.sshClient.extraParams=
```

If you have multiple IP addresses on the Web Application, external and internal, or inside a NAT network, then you need to let the DGE or DGE extension know the external (public) IP address or

domain name of the server where Web Application is running. For this create/edit the `plugin/site.properties` file and add the following line:

```
traverse.tools.sshClient.webapp.host=webapp_server_ip
```

where `webapp_server_ip` is the IP address in dotted-quad or a domain name. If there is a firewall in front of this Web Application server, it will need to allow incoming traffic on TCP/7654.

Centralized Configuration File Distribution

Configuration File

`etc/filesync.xml`

Description

By default files and directories specified by the `etc/filesync`.xml located on the system hosting the BVE server pushed out to synchronized on all DGEs and DGE extensions. Any new configuration files or changes made in these files and directories on the central BVE server are automatically distributed to all **Traverse** components within minutes. If a remote DGE or DGE-x is down when a change is made, it will update its configuration files when it reconnects to the BVE. This feature can be disabled by unchecking the File Synchronization Server option using the **Traverse Service Controller** *(page 26)*.

> **Note:** By default, customizations made in the `/plugin` directory of DGEs or DGE extensions, are not overwritten. This allows you to maintain, if you wish, separate plugin customizations for each DGE or DGE extension, unaffected by file synchronization.

Reloading Configuration Files

> **IMPORTANT:** Even though the configuration files are distributed automatically, you must reload the configuration files manually. This is done to ensure that an invalid configuration file does not impact a running system.

In order to reload configuration files or new device signatures, you can either reload the configuration files using the Web Application or else run a command line utility to reload.

Reload using Web UI

1. Log in as the `superuser` and navigate to the Superuser > **Health** tab.

   This page automatically displays which DGE or DGE extension has updated configuration files and need to be reloaded.
2. Select all DGEs with updated configuration files, and click on **Reload**.
3. Wait to see if all the components remain in the OK status and reload successfully.

Reload using Command Line Utility

Run `utils/adminUtil.pl` with the following parameters:

```
adminUtil.pl --action=reload --address=host,host --username=xyz --password=abc
```

You can specify the `--help` option for the different options.

The following files will be reloaded:

- License parameters from `etc/licenseKey.xml`
- Monitor type definition, test type definitions & application profiles from `etc/typedef/` and `plugin/monitors/`
- message handler rulesets from `etc/messages/` and `plugin/messages/`
- report definitions from under `etc/reports/`

- notification content from `etc/actions/`
- monitoring profiles from `etc/profiles/` and `plugin/profiles/`
- MIBs under `lib/mibs` and `plugin/mibs` for traps
- Plugin actions under `plugin/actions` for action profiles and Event Manager

Chapter 8

Maintenance and Disaster Recovery

In This Chapter

Overview

This chapter describes how to maintain the various **Traverse** databases and describes the tasks you need to perform to recover from problems that might occur in the system.

> **Note:** <TRAVERSE_HOME> refers to the **Traverse** installation directory.

BVE Database Maintenance

The provisioning server stores all the configuration information in an Object Oriented database called Poet FastObjects, while the DGE components use a MySQL relational database. These databases need to be backed up periodically for safety reasons, as it allows you to use the last backed up version in the event of a database corruption.

Traverse provides utilities for backup, restore, and repair of the BVE database.

In normal operating mode, the Poet database might have objects in memory and writing data to the database files randomly. Kaseya does not recommend that you back up database files while Poet is writing files to the database. The **Traverse** script described below sends special signals to the Poet database to flush all in-memory objects to disk, and allows an external backup program to copy the database files. After the backup operation completes, the script sends a signal to Poet to resume normal operation. While the backup operation is in progress, Poet continues to operate normally and caches all write transactions.

BVE Database Maintenance on Windows

Backing up the Provisioning Database (Online)

To back up the Provisioning Database while **Traverse** is operating, open a command window and execute the following commands:

```
C:
cd <TRAVERSE_HOME>\apps\poet\bin
ptxml -export -file C:\temp\provdb.xml -server localhost -base provisioning
-overwrite
```

This creates a backup of the database in `c:\temp`. You can then archive/copy the `provdb.xml` backup file to tape or other backup media.

Backing up the Provisioning Database (Offline)

To back up the Provisioning Database while offline (if **Traverse** is not operating), do one of the following:

- Copy `<TRAVERSE_HOME>\database\provisioning` and `<TRAVERSE_HOME>\database\provisioningdict` directories to a backup location.
- Export the database to a .xml file.

To create an XML export of the database while offline, execute the following commands:

1. Shut down all **Traverse** components.
2. Execute the following commands:

```
C:
cd <TRAVERSE_HOME>
utils\databaseUtil.pl --action export --file C:\temp\provdb.xml
```

This creates an exported XML copy of the Provisioning Database in `C:\temp`. Copy `provdb.xml` to a safe location.

Restoring a Copy of the Provisioning Database

To restore a copy of the Provisioning Database that was previously exported to XML, perform the following steps:

1. Shut down all **Traverse** components.
2. Execute following commands:

```
C:
cd <TRAVERSE_HOME>
utils\databaseUtil.pl --action import --file c:\temp\provdb.xml
```

If the `provisioning` and `provisioningdict` directories were copied while **Traverse** is offline, then you need to shut down **Traverse**, copy the two directories back into the `/database` folder in `<TRAVERSE_HOME>`, and start all components.

Repairing the Provisioning Database

To repair a corrupted Provisioning Database, perform the following steps:

> **Note:** Consult with **Kaseya Support** *(https://helpdesk.kaseya.com/home)* before using these commands:

1. Navigate to Start > Programs > Traverse > **Traverse Service Controller**.
2. Clear **Provisioning Database**, **Data Gathering Engine**, **Web Application and BVE API**.
3. Click **Apply** to stop the specified services.
4. Open a command window and enter:

```
cd <TRAVERSE_HOME>
apps\poet\bin\ptadmin -check database\provisioning
apps\poet\bin\ptadmin -repair database\provisioning
apps\poet\bin\ptadmin -reorg database\provisioning
```

> **Note:** If you receive any errors, contact **Kaseya Support** *(https://helpdesk.kaseya.com/home)* before attempting to restart **Traverse**.

BVE Database Maintenance on UNIX

On UNIX platforms, a backup utility is executed from the **Traverse** cron job (`<TRAVERSE_HOME>/utils/runPeriodicTasks.pl`) nightly (see **Scheduled Tasks on UNIX** *(page 77)*). By default these backup utilities create a tar-gzipped archive in the `<TRAVERSE_HOME>/database/backup` directory with names of the form `backup-mm-dd-yy,hh-mm.tar.gz`. If you want to create these files somewhere else, edit the `<TRAVERSE_HOME>/utils/db_backup.sh` script to specify the destination by changing the `backupPath` variable. Always make sure that there is sufficient disk space for the backup files.

Manually Backing up the Provisioning Database (Online)

To export the Provisioning Database to an XML file while **Traverse** is running, execute the following commands:

```
cd <TRAVERSE_HOME>/apps/poet/bin
LD_LIBRARY_PATH=<TRAVERSE_HOME>/apps/poet/lib
export LD_LIBRARY_PATH
./ptxml -export -file /tmp/proxdb.xml -server localhost -base provisioning -overwrite
```

This creates a backup of the Provisioning Database in `/tmp`. You can then archive/copy the `provdb.xml` backup to tape or other backup media.

Manually Backing up the Provisioning Database (Offline)

To back up the Provisioning Database while offline (if **Traverse** is not operating), either:

- copy `<TRAVERSE_HOME>/database/provisioning` and `<TRAVERSE_HOME>/database/provisioningdict` directories to a backup location.
- export the database to a .xml file.

To create an XML export of the database while offline, execute the following commands:

```
cd <TRAVERSE_HOME>
utils/databaseUtil.pl --action export --file /tmp/provdb.xml
```

This creates an exported XML copy of the Provisioning Database in `/tmp`. Copy `provdb.xml` to a safe location.

Restoring the Provisioning Database (Online)

To restore the Provisioning Database from a previously exported XML file, use the following commands:

```
cd <TRAVERSE_HOME>
LD_LIBRARY_PATH=<TRAVERSE_HOME>/apps/poet/lib
export LD_LIBRARY_PATH
apps/poet/bin/ptxml -import -file /tmp/provdb.xml -server localhost -base
provisioning
```

These commands assume that the exported XML file is in `/tmp/provdb.xml`. If the file is elsewhere on the system, modify the commands accordingly.

Restoring the Provisioning Database (Offline)

To restore the Provisioning Database while **Traverse** is shut down, use the `databaseUtil.pl` script:

```
cd <TRAVERSE_HOME>
etc/traverse.init stop
databaseUtil.pl --action import --file /tmp/provdb.xml
```

These commands assume that the exported XML file is in `/tmp/provdb.xml`. If the file is elsewhere on the system, modify the commands accordingly.

If you copied the `/provisioning` and `/provisioningdict` directories while **Traverse** is offline, you need to shut down **Traverse**, copy the two directories back into the `/database` folder in `<TRAVERSE_HOME>`, and start all components.

Restoring the Provisioning Database from Automated Backup

Assuming that you properly installed the **Traverse** crontab, UNIX installations of **Traverse** automatically create daily backup snapshots in the `<TRAVERSE_HOME>/database/backup` directory. Note that only a BVE host backs up the Provisioning Database. All other hosts only back up their local DGE database.

To restore the Provisioning Database from the daily backup snapshot, uncompress and un-tar the archive into the `<TRAVERSE_HOME>` directory. You must stop **Traverse** before restoring the database.

```
cd <TRAVERSE_HOME>
etc/traverse.init stop
cd database
mv provisioning provisioning.OLD
mv provisioningdict provisioningdict.old
cd ..
gunzip -c database/backup/backup-mm-dd-yy,hh-mm.tar.gz | tar xvf -
database/provisioning database/provisioningdict
<TRAVERSE_HOME>/etc/traverse.init start
```

The daily backup snapshot should include the `/provisioning` and `/provisioningdict` directories, as well as an exported XML (`provdb.xml`) copy of the database. You should use the procedure above if you have copies of the appropriate directories. If you do not have the directory snapshots, you must use the `provdb.xml` file from the backup snapshot, or the one that you created in *Manually Backing up the Provisioning Database (Online)* above.

To restore from the `provdb.xml`:

1. Shut down all **Traverse** components.
2. Execute following commands:

```
su
cd <TRAVERSE_HOME>
utils/databaseUtil.pl --action import --file /tmp/provdb.xml
```

DGE Database Maintenance

Traverse provides utilities for backup, restore, repair, and optimization of the DGE databases. You should back up the DGE databases periodically so that you can restore in the event of serious database corruption or loss.

Some common causes of DGE database corruption include the following:

- **Traverse server shuts down unexpectedly due to power failure or operating system crash.** During a normal system shutdown, the database server will flush all in-memory data to disk and properly close the database tables, but in the event of a power outage or sudden crash, the database server is not able to perform such cleanup tasks.
- **Running out of disk space on the drive or partition where Traverse is installed.** If the database server is not able to allocate disk space to add new data a table or update existing information, the corresponding table may be left in an unusable state. As a rule of thumb, available space should be three times the size of the database directory.
- **Database tables or files accessed by an external application.** This can happen on Windows when anti-virus software is configured to scan files "on access," which may corrupt database tables. You should configure anti-virus software to ignore/exclude any files in the **Traverse** database directory.

Backup software can also affect database files in a similar manner, so you should exclude the **Traverse** database directory from automated backup tasks and use only the built-in `db_backup` utility to back up the **Traverse** DGE databases.

DGE Database Maintenance On Windows

The DGE databases are located in the directory `<TRAVERSE_HOME>\database\mysql\<DB_NAME>` with each table for that database represented by a file with the extension `.MYD`. For the historical performance data collected by the DGE, the `<DB_NAME>` is `aggregateddatadb`. For the BVE there is also a database named `liveeventsdb`, which contains deduplicated events.

Traverse comes with a utility to create a fast snapshot by locking all the databases and then directly saving the raw databases.

Backing up the DGE Database

To back up the DGE database on Windows, you can manually create a backup snapshot with the following commands:

```
C:
cd <TRAVERSE_HOME>
utils\db_backup.cmd
```

This creates a new snapshot named `<TRAVERSE_HOME>\database\mysql\backup_<DB_NAME>` for each of the available databases.

Restoring the DGE Database

To restore the DGE database from a snapshot created by `db_backup.cmd`, you must restore the `.MYD` and `.FRM` files from `<TRAVERSE_HOME>\database\mysql\backup_<DB_NAME>`, and then rebuild the database indexes by performing the following steps:

1. Shut down all components using the Traverse Service Controller.
2. At a command prompt, execute the following commands, replacing <DB_NAME> with the name of the database you are restoring::

```
C:
cd <TRAVERSE_HOME>
move database\mysql\<DB_NAME> database\mysql\saved_<DB_NAME>
xcopy /E database\mysql\backup_<DB_NAME> database\mysql\<DB_NAME>\
net start nvdgedb
apps\mysql\bin\mysql --defaults-file=etc\mysql.conf --execute="SHOW TABLES"
--database="<DB_NAME>" > tables.txt
FOR /F "skip=1" %G IN (tables.txt) DO @apps\mysql\bin\mysql
--defaults-file=etc\mysql.conf --execute="REPAIR TABLE %G USE_FRM" <DB_NAME> >>
logs\database_restore.log
```

Repairing the MySQL DGE Database

Occasionally, because of an unexpected shutdown or a process such as an antivirus program scanning the database directories, database tables might become corrupt. To repair the DGE database tables, perform the following steps:

1. Shut down all components using the Traverse Service Controller.
2. Execute the following commands to rebuild the indexes:

```
C:
cd <TRAVERSE_HOME>
utils\db_repair.cmd
If the db_repair.cmd script cannot be executed or fails to repair the database (because
the extent of damage is too severe), you can manually repair the database as follows:
C:
cd <TRAVERSE_HOME>
del /f database\mysql\aggregateddatadb\*.TMD
for %f in (database\mysql\aggregateddatadb\*.MYI) do apps\mysql\bin\myisamchk
--defaults-file=etc\mysql.conf -r %f
If the -r option fails to repair the database tables, try using the -o option to perform
a slower but more effective repair method on the affected tables.
```

Optimizing the MySQL DGE Database Indexes

In some cases, the database indexes for MySQL can become inefficient and can benefit from some optimization. Generally, this is not needed. However, if the database performance suffers and it is not caused by slow disk I/O, lack of memory, or other typical causes, performing an index optimization can

improve performance. To optimize the indexes, perform the following steps:

1. Shut down all components using the Traverse Service Controller.
2. Execute the following commands:

```
C:
cd <TRAVERSE_HOME>
utils\db_optimize.cmd
```

DGE Database Maintenance on UNIX

The DGE databases are located in the directory `<TRAVERSE_HOME>/database/mysql/<DB_NAME>` with each table for that database represented by a file with the extension `.MYD`. **Traverse** comes with a utility to create a fast snapshot by locking all the databases and then directly saving the raw databases.

> **Note:** Remember to replace `<TRAVERSE_HOME>` in the commands below. Also use the correct name of your backup file where `backup-mm-dd-yy,hh-mm.tar.gz` appears in the `gunzip` command below.

Backing up the DGE Database

To back up the DGE database on a UNIX platform, create a backup snapshot with the following commands:

```
cd <TRAVERSE_HOME>
utils/db_backup.sh
```

This creates a new snapshot as a tar/gzip archive in `<TRAVERSE_HOME>/database/backup`.

Restoring the DGE Database

To restore the DGE database from a snapshot created by **db_backup.sh**, you will must restore the `.MYD` and `.FRM` files from the snapshot archive in `<TRAVERSE_HOME>/database/backup`, and then rebuild the database indexes with the following steps:

1. Shut down all components.
2. At a command prompt, execute the following commands:

```
cd <TRAVERSE_HOME>
etc/traverse.init stop
mv database/mysql/aggregateddatadb database/mysql/aggregateddatadb.OLD
gunzip -c database/backup/backup-mm-dd-yy,hh-mm.tar.gz | tar xvf -
database/mysql/backup_dge
etc/dgedb.init start restore
apps/mysql/bin/mysql --defaults-file=etc/mysql.conf --skip-column-names -u root
--password= --batch -e 'show tables;' backup_dge > /tmp/names.txt
apps/mysql/bin/mysql --defaults-file=etc/mysql.conf -u root --password= --execute
"CREATE DATABASE aggregateddatadb;"
for i in `cat /tmp/names.txt`; do apps/mysql/bin/mysql
--defaults-file=etc/mysql.conf -u root --password= --execute "RESTORE TABLE $i FROM
'<TRAVERSE_HOME>/database/mysql/backup_dge'" aggregateddatadb; done
etc/dgedb.init stop
rm -rf database/mysql/backup_dge
etc/traverse.init start
```

Repairing the MySQL DGE Database

Occasionally, because of an unexpected shutdown or a process such as an antivirus program scanning the database directories, database tables might become corrupt. To repair the DGE database tables, shut down **Traverse** and execute the following commands to rebuild the indexes:

```
cd <TRAVERSE_HOME>
etc/traverse.init stop
utils/db_repair.sh
```

If the `db_repair.sh` script cannot be executed or fails to repair the database (because the extent of damage is too severe), you can manually repair the database as follows:

```
cd <TRAVERSE_HOME>
rm -f database/mysql/aggregateddatadb/*.TMD
apps/mysql/bin/myisamchk --defaults-file=etc/mysql.conf -r
database/mysql/aggregateddatadb/*.MYI
```

If the `-r` option fails to repair the database tables, try using the `-o` option to perform a slower but more effective repair method on the affected tables.

Optimizing the MySQL DGE Database Indexes

In some cases, the database indexes for MySQL can become inefficient and can benefit from some optimization. Generally, this is not needed. However, if the database performance suffers and it is not caused by slow disk I/O, lack of memory, or other typical causes, performing an index optimization can improve performance. To optimize the indexes, use the following commands:

```
cd <TRAVERSE_HOME>
utils/db_optimize.sh
```

Switching to a Backup DGE

The following steps describe how to switch to a backup DGE if a DGE in your **Traverse** environment fails.

Switching to a Backup DGE

1. Edit the `dge.xml` file of the backup DGE so that the name of the backup DGE is the same as the DGE that failed.
2. Log in to the Web Application to which the failed DGE is associated.
3. Change the IP address of the failed DGE to the IP address of the backup DGE.
4. Restart the DGE service.
5. Restart the Web Application in the BVE.

Moving Traverse from UNIX to Windows

The following description is for transferring an existing copy of **Traverse** from a UNIX platform to a Windows platform.

Transferring Traverse from UNIX Platform to Windows Platform

1. Install (fresh) **Traverse** on the Windows servers. Make sure that the Windows host is restarted after the installation is complete. When the server restarts, shut down **Traverse**, which starts automatically, using the Service Controller. You must also shut down the UNIX host using `etc/traverse.init` stop.
2. Export the Provisioning Database from UNIX host using following commands Substitute proper path names if required.

```
cd <TRAVERSE_HOME>
utils/databaseUtil.pl --action export --file /tmp/provdb.xml
```

3. When the command completes, copy `/tmp/provdb.xml` to a temporary location on the Windows host.

4. Import the data into the Provisioning Database on Windows host by opening a command window and executing the following commands:

```
c:
cd <TRAVERSE_HOME>
utils\databaseUtil.pl --action import --file C:\temp\provdb.xml
```

   Substitute the correct path if `provdb.xml` is in a location other than `c:\temp`. When prompted to proceed with this operation, enter `y`. The command takes a few minutes depending on how many objects are configured in the database from the UNIX host.

5. For transferring the DGE database, you can create a zip archive of `<TRAVERSE_HOME>/database/mysql/aggregateddatadb` directory and extract the files under the same directory on the Windows host.
6. Copy your permanent license (`etc/licenseKey.xml`) from the UNIX host to the Windows host.

> **Note:** Step 2 and Step 3 are only applicable for migrating the host that is running BVE/Web Application. Likewise, Step 4 is only for migrating the DGE host.

Password Recovery

Recovering a Password on Windows

1. Stop all **Traverse** components (Start > Programs > Traverse > Stop Traverse Components).
2. Open a command prompt and execute the following commands:

```
c:
cd <TRAVERSE_HOME>
utils\databaseUtil.pl --action export --file provdb.xml
```

   This creates a file named `provdb.xml` which you can edit to reset the password.

3. Search for the `superuser` entry and remove the entire `des` prefix along with the encrypted password. Then, enter a cleartext password (with no `des` prefix).

```
<loginName code="Ansi" lang="en">superuser</loginName>
[...]
<password code="Ansi" lang="en">{des}xyzabc</password>
```

   replace with:

```
<password code="Ansi" lang="en">password</password>
```

4. Import the file back into the database:

```
utils\databaseUtil.pl --action import --file provdb.xml
```

5. Enter `y` to replace the existing database.
6. Restart **Traverse**.

Recovering a Password on UNIX

1. Log in to the Traverse (BVE) server as root or use the `su` or `sudo` commands to obtain root permissions.
2. Execute the following commands:

```
cd <TRAVERSE_HOME>/
/etc/traverse.init stop
utils/databaseUtil.pl --action export --file /tmp/provdb.xml
```

   This stops the **Traverse** components, and then creates a file named `provdb.xml` which you can edit to reset the password.

3. Search for the `superuser` entry and remove the entire `des` prefix along with the encrypted password. Then, enter a cleartext password (with no `des` prefix).

```
<loginName code="Ansi" lang="en">superuser</loginName>
[...]
<password code="Ansi" lang="en">{des}xyzabc</password>
```

replace with:

```
<password code="Ansi" lang="en">password</password>
```

4. Import the file back into the database:

```
utils/databaseUtil.pl --action import --file /tmp/provdb.xml
```

5. Enter `y` to replace the existing database.
6. Restart **Traverse**.

Expiring Messages

If you want manually expire old event messages from the database for a device, perform the following steps:

> **Warning:** Use extreme caution when issuing SQL delete statements. Kaseya recommends that you have a current backup of the database and that you verify each command before execution. Failure to take proper precautions can result in corruption or lost data.

Expiring Messages on Windows

1. Open the message windows and record the **Device Address** for the device.
2. Log in to each **Traverse** DGE and execute the following commands:

```
C:
cd <TRAVERSE_HOME>
apps\mysql\bin\mysql -u root --password= aggregateddatadb
(then, at the mysql> prompt)
UPDATE ALARMS set expireTime=1081262436203 WHERE deviceAddress='n.n.n.n' and
expireTime=-1;'
quit;
where n.n.n.n is the IP address. When the database table updates, the messages no longer
display in the message window.
```

Expiring Messages on UNIX

1. Open the message windows and record the **Device Address** for the device.
2. Then log in to each **Traverse** host and execute the following commands:

```
cd <TRAVERSE_HOME>
etc/dgedb.init admin dge
```

3. Then, at `mysql>` prompt enter

```
UPDATE ALARMS set expireTime=1081262436203 WHERE deviceAddress='n.n.n.n' and
expireTime=-1;'
quit;
```

where `n.n.n.n` is the IP address. When the database table updates, the messages no longer display in the message window.

Changing the IP Address of the BVE

Because the Provisioning Database stores all device and test parameters, aggregation scheme, test schedules, action profiles, and such, the DGE component must have the IP address of the host on

which the Provisioning Database is operating.

Similarly, because the database contains information about user accounts, various limits and permissions, service definitions, and such, the Web Application component must communicate with the Provisioning Database on a regular basis.

Therefore, to change the IP Address of the BVE, do the following steps:

Changing the IP Address of the BVE

1. On the BVE, open `etc/emerald.xml`.
2. Locate the following section:

```
<provisioning name="provisioning"
host="n.n.n.n"
[....]
```

   and change the old IP address (`n.n.n.n`) to the new IP address of the BVE. Also configure the JMS server by updating the IP address in the following section of the above file:

```
<jms host="n.n.n.n"
[....]
```

3. Open `etc/openjms-mysql.xml`. Locate and update the IP address in the following section of this file:

```
<ServerConfiguration host="n.n.n.n"
embeddedJNDI="true" />
```

4. Edit `etc/emerald.properties` and update the `org.quartz.dataSource.myDS.URL` section.

```
org.quartz.dataSource.myDS.URL=jdbc://mysql://n.n.n.n:7663/schedulerdb
```

5. (UNIX) Edit `etc/emerald.env` and update the `OPENJMS_HOST` variable:

```
OPENJMS_HOST="n.n.n.n"
```

6. On the DGE, edit `etc/emerald.xml` as you did in Step 1 and Step 2.

Scheduled Tasks on UNIX

Traverse provides a sample crontab file that contains periodic maintenance tasks to ensure the proper operation of the **Traverse** system. The contents of this file should be added to root's crontab: `<TRAVERSE_HOME>/etc/emerald.crontab`.

Chapter 9

APPENDIX B: Troubleshooting Traverse

In This Chapter

General Troubleshooting Information

This section includes general troubleshooting information for both Windows and UNIX operating systems.

Log Files

Several log files can be useful in troubleshooting. All log files are located under `<TRAVERSE_HOME>\logs` directory.

Log File	Used By
`stderr.log`	All startup scripts, monitors
`error.log`	Any warning, error or critical level messages generated by the application are logged in this file.
`monitor.info`	Information on monitors are logged to this file as tests are performed, actions triggered, etc.
`webapp.info`	All user tasks, both in the Web Application and BVE socket server are logged to this file. Tasks include create, delete, update, suspend and resume tasks performed on devices, departments, users, etc.
`tomcat.log`	Any errors generated inside JSP pages in the Web Application component is logged in this file.
`poet.log`	Provisioning Database specific errors

Troubleshooting the DGE-BVE Connection

Upon startup, each DGE component connects to the Provisioning Database located on the provisioning server and downloads all tests that are configured for that DGE. The DGE components maintain a connection to the Provisioning Database at all times. As devices and tests are added, updated, or removed, the provisioning server notifies the relevant DGE of the changes in real time.

If the communications link between the Provisioning Database and the DGE is broken, the DGE repeatedly attempts to restore the connection, while continuing to monitor, using the configuration information that it has cached in memory. Once the connection to the Provisioning Database is restored, the DGE shuts down. A cron job restarts the DGE shortly thereafter. The reason for the shutdown and restart is that while the DGE was unable to communicate with the provisioning server, it may have missed notices about changes to device/test configurations. In the process of restarting, the DGE downloads a fresh copy of the list of tests and proceeds with normal operation.

Querying SNMP Devices Manually

To query SNMP devices manually, execute the following commands:

- Windows

```
cd <TRAVERSE_HOME>
bin\snmpwalk -m "" -c public -v 2c ipAddress:port mib
```

Example

```
bin\snmpwalk -m "" -c public -v 2c 10.1.2.3 .1.3.6.1.2
```

- UNIX

```
cd $TRAVERSE_DIR
bin/snmpwalk -m "" -c public -v 2c ipaddr:port mib
```

Frequently Asked Questions and other Problems

The section addresses FAQs and various other issues that might occur while using **Traverse**.

What changes are required when I change IP address of a DGE host?

When the IP address of the DGE host is changed, no configuration change is required on the DGE itself. However, the BVE needs to know the IP address of the DGEs in order to query them for the reporting data.

You will need to update the new IP address in the BVE. See **Configuring a New DGE** *(page 30)*.

What changes are required when I change the IP address of the BVE database server?

The DGEs need to know the IP address of the Provisioning Database to download their configuration. Furthermore, the JMS Messaging server also runs on this host and so its address must be updated on the DGEs.

On all the DGEs, edit `<TRAVERSE_HOME>\etc\emerald.xml` file and update the following section:

```
<provisioning name="provisioning"
host="n.n.n.n"
[...]
```

as well as the JMS section:

```
<jms
host="n.n.n.n"
[...]
```

On the host running the Provisioning Database, edit `<TRAVERSE_HOME>\etc\openjms-mysql.xml` and update the following section:

```
<ServerConfiguration host="n.n.n.n" embeddedJNDI="true" />
```

Then restart the **Traverse** components on all the servers.

Can I use a different TCP port for MySQL? (Unix)

In order to change the port used by the aggregated database (MySQL), complete the installation of **Traverse** and then do the following steps:

Windows

1. Stop **Traverse** if it is already running using the controller.
2. Edit `<TRAVERSE_HOME>\etc\my.ini` and change the port number specified by `port=nnnn` entries. There should be two such entries and you should specify the same value for both.
3. Edit the configuration file `TRAVERSE_HOME\etc\emerald.xml` and locate the following section:

```
<dge vendor="mysql"
port="nnnn"
[...]
```

   Change the value of the port parameter to the new port number entered in `my.ini` above.
4. Restart **Traverse**.

UNIX

1. Stop **Traverse** if it is already running using `TRAVERSE_HOME/etc/traverse.init`.
2. Edit the configuration file `<TRAVERSE_HOME>/etc/mysql.conf` and change the port number specified by `port=nnnn` entries. There should be two such entries, and you should specify the same value for both.
3. Edit `TRAVERSE_HOME/etc/emerald.xml` and locate the following section:

```
<dge vendor="mysql"
port="nnnn"
[...]
```

   Change the value of the port parameter to the new port number entered in `mysql.conf` above.

4. Edit `TRAVERSE_HOME/etc/emerald.env` and locate the following section:

```
MYSQL_PORT="nnnn"
```

   Change the value to the new port number entered in mysql.conf.

5. Restart **Traverse**.

Can I run the Web Application on a different TCP port?

See **Web Server TCP/IP Port** *(page 62)*.

How do I enable SSL support on the Web Application?

Since the **Traverse** UI is pure HTML based, you can access it using http or https. See **Configuring SSL for the Web Application** *(page 19)* for more information on setting this up.

DGE does not automatically restart when the connection to the Provisioning Database is restored

Make sure that the crontab entry for root on the DGE includes the contents of `<TRAVERSE_HOME>/etc/emerald.crontab`.

Chapter 10

APPENDIX F: NCM Requirements

In This Chapter

Enabling the NCM Module on Unix

If your **Traverse** installation is on a UNIX server, make sure you have the following Perl modules installed before enabling NCM:

- Archive::Tar
- Compress::Raw::Zlib
- Compress::Raw::Bzip2
- Crypt::SSLeay
- Crypt::DES
- Digest::SHA1
- IO::Compress::Base
- List::Util
- Math::BigInt::GMP
- MIME::Base64
- Socket6
- Term::ReadKey
- Time::HiRes
- XML::Parser

In most Linux installations, you can run the following command as root to install the required Perl modules:

Installing Perl Modules

```
yum install perl-Crypt-SSLeay perl-Crypt-DES \
  perl-Digest-SHA1 perl-List-Util perl-XML-Parser \
  perl-Socket6 perl-Time-HiRes perl-Math-BigInt-GMP \
  perl-MIME-Base64 perl-IO-Compress-Base \
  perl-Compress-Raw-Zlib perl-Compress-Raw-Bzip2 \
  perl-Archive-Tar perl-Term-ReadKey
```

Enabling NCM

- Edit `<TRAVERSE_HOME>/etc/emerald.init` and locate the following line: `NETCONF="N"`
- Change the `"N"` to `"Y"` so that the NCM components are started automatically along with the other **Traverse** components.