

Chapter 3

Configuration and Operations

3.1 Overview

This document provides step-by-step instructions on how to configure and run Helix on your production environment. It is assumed that the software package has been extracted and installed into a specific directory on your server(s) as described in Chapter 2, “Installation and Upgrade Guide”.

NOTE: In this document, `HELIX_HOME` refers to the top level directory where the package has been extracted and will be running from.

3.2 Installing A New License Key

Fidelia Helix relies on a license key to indicate which features are available, and also to impose time restrictions (if any) on when the application “expires”. It may be necessary to install a new license key, for example, when a permanent license key is provided by Fidelia at the end of a trial period, or when the key format changes between different versions of the application. Once a new key has been delivered, here are the steps you need to take to install the new key:

□ To install a new license key:

1. Save/copy the downloaded file as `HELIX_HOME/etc/licenseKey.xml`. On Linux/Solaris platforms this is `/usr/local/helix` by default, and on Windows platforms this directory is `\Program Files\Fidelia Helix`.
2. Make sure to replace the contents of the existing `licenseKey.xml` with this new file.

- Restart Helix using `HELIX_HOME/etc/helix.init restart` on Linux/Solaris platforms, or `Start | Programs | Fidelia Helix | Start Fidelia Helix` on Windows platforms.

3.3 Starting/Stopping Helix

On Linux/Solaris platforms, Helix is started and stopped using the `HELIX_HOME/etc/helix.init` script. This script should be called from `/etc/rc.local` or other startup directory appropriate to your operating system with a parameter of `start` so that Helix components start automatically when the system reboots. On Windows, Helix is installed as a Windows service, and should be started using `Start | Programs | Fidelia Helix | Start Fidelia Helix`.

Although under normal circumstances you would run the `helix.init` script or menu items from Program Files, each component of Helix system has its own startup script in-case you would like to start/stop any of the components individually. These scripts are located under the `HELIX_HOME/etc` directory on Linux/Solaris platforms. On the Windows platform, use `net start <service_name>` and `net stop <service_name>`. The scripts are named:

Table 3.1

Script Name	Windows Service	Description
<code>provdb.init</code>	<code>nvprovdb</code>	provisioning server/database (poet)
<code>dgedb.init</code>	<code>nvdgedb</code>	DGE/monitor database (mysql)
<code>monitor.init</code>	<code>nvmonitor</code>	DGE/monitors
<code>webapp.init</code>	<code>nvwebapp</code>	web interface
<code>bveapi.init</code>	<code>nvbveapi</code>	BVE API server

Each of these scripts accepts the parameters `start` and `stop` which will start and stop the respective component.

3.3.1 Starting the system

The provisioning database starts up first since all other components request configuration information from the provisioning database. The DGE database, monitors and web application follow next. The `helix.init` script on Linux/Solaris platforms will take care of maintaining this order.

```
% /etc/init.d/helix.init start
```

On Windows platforms, select `Start | Programs | Fidelia Helix | Start Fidelia Helix` to start the entire application.

3.3.2 Stopping the system

On Linux/Solaris use the following command to stop Helix:

```
% /etc/init.d/helix.init stop
```

and on Windows, `Start | Programs | Fidelia Helix | Stop Fidelia Helix`.

If you want to stop the components of Helix that read configuration files (to re-read these config files), then you can also use

```
% /etc/init.d/helix.init stopcore
```

This will not stop the various databases or the messaging bus.

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 configuration database was unable to start up properly and you should try again after a few seconds.*

3.3.3 Verifying proper operation

On Linux/Solaris platforms, using the `status` parameter with the `helix.init` script will display the status of the different components. Example:

```
% ./helix.init status
```

```
messaging server (openjms) ... running
provisioning database (poet) ... running
dge (monitor) components ... running
dge/jms database (mysql) ... running
application server (tomcat) ... running
virtual frame buffer (xvfb) ... running
```

On Windows platform, you can check the status of individual components using the Service Control Manager where the Status column should indicate `Started` when a particular component is running. You can also execute `net start | more` from a command prompt to get a list of running services. Helix components are prefixed with “Helix”.

3.4 Configuration Files

Helix system utilizes 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 following files:

3.4.1 Application installation path

Configuration File	Affected Components	Affected Operating Systems
<code>HELIX_HOME/etc/helix.env</code>	Provisioning database, Web application, Monitor	Linux, Solaris

This file contains environment variables that specify the location of different supporting software needed to run Helix.

`INSTALL_DIR` should be set to the installation directory, `HELIX_HOME` (as described above). All other variables should be left unchanged unless specified otherwise by Fidelia support.

3.4.2 Logging configuration

Configuration File	Affected Components	Affected Operating Systems
HELIX_HOME/etc/log4j.conf	Provisioning database, Web application, Monitor	Linux, Solaris, Windows

Different components of Helix provide useful diagnostic and/or informative log messages, and you can control how much information is logged by editing this file. Change `LOGLEVEL` to one of the following to fine tune the level of details you would like:

Table 3.2 Log message detail levels

LOGLEVEL	Level of Detail
INFO	Informational messages that highlight the progress of the application at 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 Fidelia technical support.

By default, messages are only logged into Helix's own log files stored in the directory specified by `$LOGDIR` variable. If you would like to send the logs to a Unix syslog host, either at a central location, or on same host(s), 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
```

and change `localhost` to the fqdn or ip address of the host where you want the log messages to be sent. If you would like the messages to be sent as a facility other than `local7`, change

LOG_LOCAL7 to LOG_<FACILITY> where <FACILITY> is one of the facilities listed in the Unix manual (man5) of `syslogd.conf`. Make sure to enter the facility name in upper case.

3.4.3 Web application external help

Configuration File	Affected Components	Affected Operating Systems
HELIX_HOME/webapp/WEB-INF/web.xml	Web application	Linux, Solaris, Windows

Helix provides an easy way to add escalation information, procedures or any other information related to individual tests, or on a global basis on test type, device, or Department context. Each test item on the web application includes a **HELP** link, which when clicked on, shows any such information. This information is obtained by running an external script. Locate the section containing:

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

This parameter identifies the location of this script. Helix includes a default script, located at

HELIX_HOME/Utils/externalTestHelp.pl, which looks for such information in a directory hierarchy of specific layout.

If you would like to have a different script used for this feature, you can change the `externalTestHelp.pl` script name and path to specify the different script.

For the algorithm that is used to find test-specific information, see Section 5.3, “External Help” on page 56.

3.4.4 DGE controller port/password

Configuration File	Affected Components	Affected Operating Systems
HELIX_HOME/etc/dge.xml	Monitor	Linux, Solaris, Windows

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 into the status server.

3.4.5 E-mail servers

Configuration File	Affected Components	Affected Operating Systems
HELIX_HOME/etc/helix.xml	Monitor, Report Server	Linux, Solaris, Windows

The DGE and Report Server components need to know which E-mail server(s) they should use to send notifications or reports via E-mail. Edit the following section:

```
<email-servers>
<host name="my_mail_server" priority="10"/>
</email-servers>
```

Change `my_mail_server` to the fqdn of your local E-mail server or the E-mail server that you use for sending outgoing E-mail. If you have more than one E-mail server, you may add additional servers with a different priority value. The Helix component responsible for sending mail will start with the E-mail server with the lowest priority, and if it is unable to reach that server, it will move onto the next server on the list until the notification has been sent out successfully. You should make sure that the E-mail

server(s) is configured properly to allow Helix to relay E-mail to any E-mail address. (Please refer to your E-mail server's administration guide for instructions on how to accomplish this.)

3.4.6 Web server TCP/IP port

Configuration File	Affected Components	Affected Operating Systems
HELIX_HOME/tomcat/conf/server.xml	Web application	Linux, Solaris, Windows

This is the configuration file for Jakarta Tomcat application server. If you would like to run the application on a different port other than the default of port 80, you need to locate the section:

```
<Connector
  className="org.apache.tomcat.service.PoolTcpConnector>
```

and change the value of <Parameter name="port">.

3.4.7 Web User Interface Appearance

Configuration File	Affected Components	Affected Operating Systems
HELIX_HOME/webapp/resources/css/styleSheet.css	Web application	Linux, Solaris, Windows

This is an HTML-standard compliant style sheet that is used by the Helix web interface. You can edit the definitions of different HTML components in this file to change the look and feel of the web interface to suit your needs. This stylesheet is used only on those sections of the interface that are visible to the user.

3.5 Network Device Discovery

Today, enterprise networks are large, complex, and constantly changing. To help you keep track of the components of your infrastructure, Helix provides device discovery to automatically find available devices on your network and determine available tests on the devices.

Additionally, you can also map topology relationships between devices by creating device dependencies as described in Section 5.2, “Device Dependency” on page 54.)

NOTE: You must log in as “admin” to run network discovery.

3.5.1 Running Discovery

To discover the devices on your network, log in as ‘admin’ and then go to the Manage > Discovery menu.

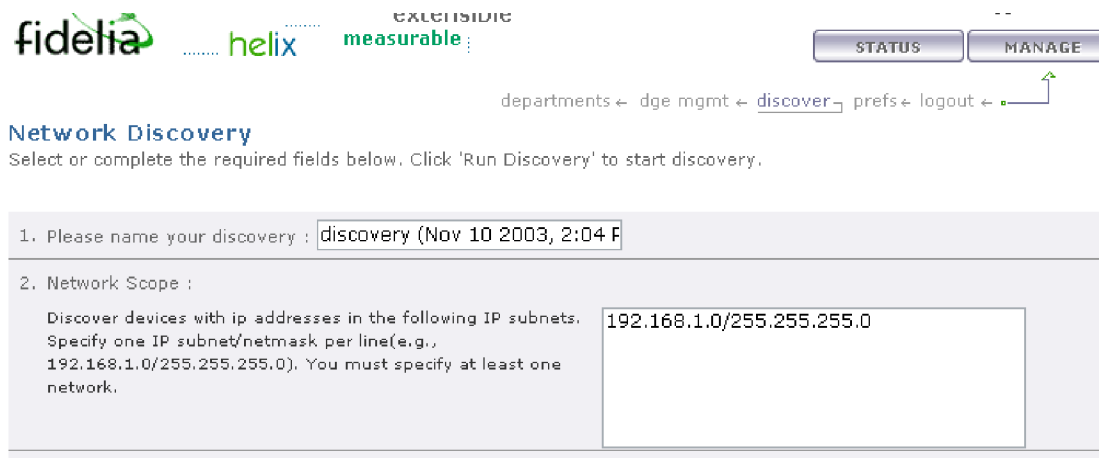


Figure 3.1 Network Discovery screen

1. If no discovery data exists, the Network Discovery page appears. Configure the following discovery options:

Table 3.3 Network Discovery Configuration Parameters

Field	Purpose
Network Discovery Scope	Specify the subnet(s) on which you want to discover devices. For each subnet, enter an IP address and a subnet mask in dotted quad notation. To enter multiple IP address/subnet mask pairs, list each one on a separate line. You must specify at least one subnet/subnet mask pair. For additional information see Section 3.5.1, “Running Discovery” on page 27.
Discovery Location	Do not change.
SNMP Community Strings	To automatically discover SNMP tests that are supported by discovered devices, specify the SNMP community strings that are used in your network(s). Enter one community string per line. If no community string is entered, discovered devices are not tested for SNMP capabilities To discover SNMP devices only, select Exclude devices that do not support SNMP . If this option is selected you must enter at least one SNMP Community string.

4. SNMP Community Strings :

Exclude devices that do not support SNMP. If selected, at least one SNMP community string must be specified here. To automatically discover SNMP tests that are supported by discovered devices, specify the SNMP community strings that are used in your network(s). Enter one community string per line. If no community string is entered, discovered devices are not tested for SNMP capabilities.

public

Run Discovery Reset

2. Click **Run Discovery**. While discovery is in progress, the Network Discovery Status page periodically refreshes the status display. Discovery may take several minutes to complete, up to several hours in large networks with tens of thousands of devices.
3. If devices are discovered, the Network Discovery Results page displays them, sorted by device type. (Devices with an unrecognized type are listed as **Type: Unknown/Other**.) To provision discovered devices and matching tests discovered, select the **Department** to which you want to assign the devices and the devices that you want to provision, and then click **Provision**.

After the operation is complete, the Network Discovery Status window displays a message indicating that the devices were successfully provisioned. For each provisioned device, Helix creates ICMP ping tests (packet loss and RTT).

NOTE: Devices that are already provisioned (with the same name) are not created again.

3.6 Windows monitoring using WMI

3.6.1 Overview

Helix can monitor Windows hosts using the native Windows Management Instrumentation (WMI), which is installed by default on all Windows 2000, XP and 2003 or later versions, and available as an add-on for Windows NT hosts.

Helix performs WMI monitoring using the Helix WMI Query Server (nvwmisd). This server is automatically installed on Windows DGEs, however, to perform WMI monitoring from a

Linux or Solaris DGE, you must install and configure a Helix WMI Query Server as a “proxy” on a Windows system that can access the Windows hosts to be monitored.

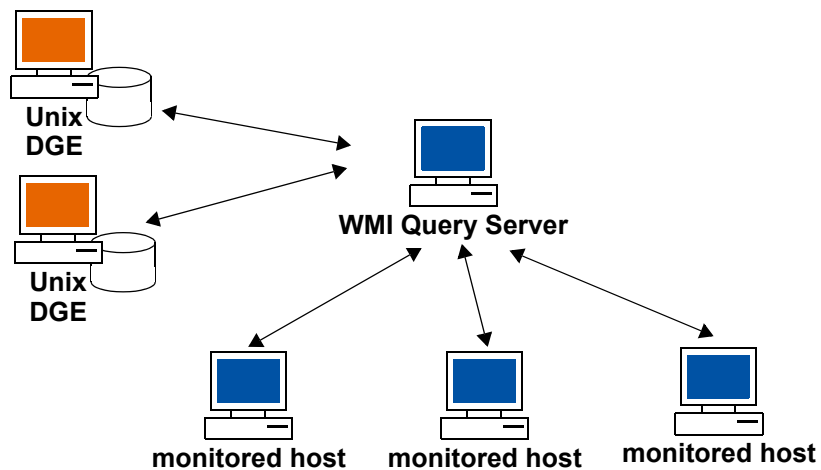


Figure 3.2 Relationship between DGEs, WMI Query Server, and monitored hosts

3.6.2 Installing the Helix WMI Query Server

NOTE: This service is automatically installed on Windows DGEs by default and is only needed if your Helix installation is on a Unix/Linux server.

The WMI Query Server (nvwmisd) should be installed on a Windows machine which has access to the windows hosts being monitored using NetBIOS. Test this by typing `NET VIEW \\remote_host` at a Windows Command Prompt.

Access requirements

The WMI Query Server queries all windows hosts using a common username and password for ease of management, instead of a separate user for each host. So you will need to ensure that each Windows host to be monitored through WMI has a user account that can be accessed by the WMI Query Server (with

administrative rights to access various system tables). This username and password are stored in cleartext in the `nvwmigd.ini` configuration file in the `HELIX_ROOT/etc/` directory.

- If the Windows hosts 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

System Requirements

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

- Pentium III processor, 512MB RAM, 10MB free disk space
- Windows 2000 with SP3 or Windows XP with SP1

Firewall requirements

- When used in a ‘proxy’ mode, the DGE communicates with the WMI Query Server on TCP port 7667 by default. To specify a different port number, edit the configuration files on the Query Server and DGE as described in “Operating Helix Behind Firewalls” on page 46 and Section 3.11, “Operating Helix Behind Firewalls” on page 46. If firewalls, access lists, etc. exist between the DGE and the WMI Query Server, the rules must be modified to allow incoming connections from the DGE to the WMI Query Server on the specified port. The rules should allow persistent connections (i.e., connections should not be forcibly timed out, even if there is no data flowing).

Installing the WMI Query Server

1. Download the WMI Query Server (wmiQDinstaller.exe) from the Helix CD-ROM or the Fidelia Support site (<http://www.fidelia.com/helix/support>).
2. Double-click the install file, `wmiQDinstaller.exe`
3. Read the Introduction, and then click **Next** to continue.
4. Optionally, in the Choose Install Folder window, specify the folder in which you want to install the WMI Query Server. Click **Next** to continue.
5. In the Remote Query Credentials window, enter the username and password that the WMI Query Server will use to access monitored Windows hosts. The username can include a domain name (e.g., `ACMECORP\wmi_user`). In the **Password (Again)** field, enter the password a second time for validation. If you do not enter a **Remote Username** and **Remote Password**, the WMI Query Server will use the username and password of its local system account.
This username and password are stored in clear text in the configuration file, so this file should be protected from general user access. It is strongly recommended that a separate user account be setup on all hosts being monitored using WMI.
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.

The Helix WMI Query Server Configuration File

.....

After the Helix WMI Query Server is installed, either on a stand-alone server to be used with DGE on a Linux/Solaris server, or as part of the DGE on a Windows server, you can fine-tune it's configuration. The table below lists the configurable parameters in the configuration file `nvwmigd.ini`. This file can be found in the

WMI_Query_Server_Install_Dir\bin (stand-alone install), or HELIX_HOME\etc. If the file, or any of the parameters are missing, default values are used by the server.

NOTE: The configuration file may contain parameters that are not listed here. Do not modify unlisted parameters unless advised to do so by Fidelia technical support.

Table 3.4

Parameter	Description	Default Value
Port	TCP port on which Helix WMI Query Server listens for incoming connections from DGEs	7667
Username	User name that a DGE must use when logging in to Helix WMI Query Server	wmiuser
Password	Password that a DGE must use when logging in to Helix WMI Query Server	fixme
Server_Username	User name that the Helix WMI Query Server uses to connect to Windows hosts being monitored	n/a
Server_Password	Password that the Helix WMI Query Server uses to connect to the Windows hosts being monitored.	n/a

Example: Sample Configuration File (nvwmisd.ini)

```

.....
[ServerConfig]
Port = 7667
Username = dgeuser
Password = fixme
Timeout = 100000
Threads = 4
Server_Username = ACMECORP\localuser
Server_Password = testpassword
.....

```

3.6.3 DGE Configuration for Proxy WMI Server

If you have any Unix/Linux DGEs which need to use the WMI Query Server on a Windows machine as a proxy, edit the following parameters in `$HELIX_HOME/etc/dge.xml`:

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

and restart the DGE so that the changes can take effect.

The various parameters in the `dge.xml` file are:

host name	is a unique, descriptive name for the WMI Query Server host that this DGE uses for WMI monitoring (e.g., <code>Denver_WMI_QueryHost</code>).
address	is 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 <code>127.0.0.1</code>
port	is the TCP port on the WMI Query Server to which the DGE connects. This must match the <code>Port</code> parameter in the <code>nvwmiqd.ini</code> file on the WMI Query Server.
username	is the username that the DGE uses to log in to the WMI Query Server. This must match the <code>Username</code> parameter in the <code>nvwmiqd.ini</code> file on the WMI Query Server.
password	is the password that the DGE uses to log in to the WMI Query Server. This must match the <code>Password</code> parameter in the <code>nvwmiqd.ini</code> file on the WMI Query Server.

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

3.6.4 Provision WMI Hosts in Helix

Once you have the WMI Query Server installed, adding hosts to be monitored using WMI is done similar to other devices. You need to ensure that the devices are added to a location where all the DGEs are 'WMI Enabled' (have access to a Helix WMI Query Server).

1. Provision the Windows hosts that will be monitored using WMI as described in Section 9.1, "Managing Devices" on page 83.
2. Discover and provision WMI tests on the Windows hosts as described in Section 9.2, "Managing Standard Tests" on page 88.

3.6.5 Troubleshooting

This section lists problems that may arise with WMI monitoring, possible causes, and solutions.

- ❑ **Problem: "The DGE can't discover WMI tests for Windows hosts."**
 - Ensure that the Helix WMI Query Server is running
 - Verify that the username and password being used by the WMI Query server (in nvwmisd.ini) is for a valid administrator account on the target hosts.
 - Check the error log on the WMI Query server for errors.

- ❑ **Problem: "The DGE discovered WMI tests, but it can't monitor configured tests using WMI."**
 - Ensure that the Helix WMI Query Server is running
 - Check the error log on the WMI Query server for any errors.

❑ **Problem: “Ever since I installed WMI, I’m getting test provisioning errors.”**

- When you use Helix to discover tests, it may discover SNMP and WMI tests with the same name. However, if you try to create SNMP and WMI tests with the same, you will get provisioning errors. To keep names unique, use a naming convention to distinguish between SNMP and WMI tests. For example, start the names of all WMI tests with “wmi_”.

3.7 Setting up Apache Monitor

The apache server will need to be compiled with mod_status support. By default this module is included in a build process, but you can verify this using the following commands:

```
(on Unix)
cd /path/to/apache
bin/httpd -l | grep mod_status
```

```
(on Windows)
cd \path\to\apache
bin\httpd -l | findstr "mod_status"
```

If the output shows “mod_status.c” then this module is included in the web server. You will need to enable this module in “httpd.conf”. Refer to the documentation at http://httpd.apache.org/docs/mod/mod_status.html for instructions on how to enable and configure it. You will need to make sure that:

1. The URL for the module matches the URI specified for the Helix Apache monitor specified in the configuration file (described below).
2. “ExtendedStatus On” directive is applied
3. Helix hosts (WebApp, DGE) are included in the “Allow” directive

For example, if the Web Application is running on host 192.168.100.5, and the DGE is on host with address 192.168.200.10, the module configuration in httpd.conf may look like:

```
<Location /server-status>
  SetHandler server-status
  ExtendedStatus On
  Order Deny,Allow
  Deny from all
  Allow from 192.168.100.5
  Allow from 192.168.200.10
</Location>
```

Once the configuration has been enabled/modified, the httpd process will need to be restarted (<http://httpd.apache.org/docs/stopping.html>) to apply the changes.

Editing Helix Configuration

Once Helix has been installed, on each host (if running on a distributed configuration) edit "HELIX_HOME/lib/ext/gp_apache/config.xml" and locate the following section:

```
<uri>
  <protocol>http</protocol>
  <port></port>
  <file>/server-status?auto</file>
</uri>
```

The item of interest here is the URI specified within the <file>...</file> property. When a client access this URL, Apache server provides performance statistics that it stores in an internal database. The URL (/server-status) should match the URL specified in the configuration of the "mod_status" module (see below). If the server is configured to provide the performance stats via URL "/internal/stats/apache", then the configuration above would be changed to:

```
<uri>
  <protocol>http</protocol>
  <port></port>
  <file>/internal/stats/apache?auto</file>
</uri>
```

Note that the "?auto" parameter should be left as-is. The remaining configuration items should not be altered in any way without specific instructions from Fidelia Support.

3.8 Alphanumeric Paging

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

□ To configure Helix for alphanumeric paging:

1. Add modem configuration information to `HELIX_HOME/etc/helix.xml` on the DGE that will send the paging notifications. See Section 3.8.1, “Modem Configuration” on page 38 for details.
2. On the same DGE, add paging central information for the paging service provider to `HELIX_HOME/etc/helix.xml` as described in Section 3.8.2, “Paging Central Configuration” on page 40.
3. Create Action Profiles that use alphanumeric paging as described in Section 9.6, “Managing Action Profiles” on page 107, and assign them to tests that are run by this DGE.

3.8.1 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 `HELIX_HOME/etc/helix.xml` file. If multiple modems are configured, they are used in the order specified by their `device priority` parameters (the lower the number, the higher the priority). For each modem, set the following:

Parameter	Purpose
<code>sender id</code>	The phone number used to identify this modem when sending a page. You can set it to any phone number representing this DGE.
<code>device priority</code>	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, Helix uses the highest-priority modem that is available.

Parameter	Purpose
port	The port through which this modem communicates. For UNIX DGEs enter a port in the format /dev/ttyS <i>n</i> where <i>n</i> is 0,1,2. For Windows DGEs use the format COM <i>n</i> where <i>n</i> 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.

Sample helix.xml modem configuration

```

.....
<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>
.....

```

3.8.2 Paging Central 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 `HELIX_HOME/etc/helix.xml` file. For each service provider, set the following:

Parameter	Purpose
<code>name</code>	A name that uniquely identifies this service provider to the DGE.
<code>number</code>	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 or similar sites.
<code>speed</code>	The highest speed supported by the service provider. Possible values are 0 (110bps), 2 (300bps), 4 (1200bps), 5 (2400bps), 6 (4800bps), and 7 (9600bps). Default value is 5.
<code>parity</code>	The type of parity checking, if any, supported by the service provider. Possible values are 0 (none), 1 (odd), 2 (even), 3 (mark), and 4 (space). Default value is 2.
<code>databits</code>	The number of data bits supported by the service provider. Possible values are 2 (7 bits) and 3 (8 bits). Default value is 2.
<code>stopbits</code>	The number of end-of-byte bits supported by the service provider. Possible values are 1, 1.5, and 2. Default value is 1.
<code>flowcontrol</code>	The type of handshaking supported by the service provider to prevent data loss during transmission. Possible values are 0 (none), 1 (XONXOFF), 2 (CTSRTS), and 3 (DSRDTR). 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 location(s), 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. Undelivered pages older than 1 hour are ignored (these parameters can be controlled via the configuration in `helix.xml` also).

Sample `helix.xml` 'paging central' configuration

```

<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=ctsdrtr, 3=dsrdtr -->
  </paging-central>
  <paging-central name="nextel"> <!-- name should be unique -->
    <number>3035551212</number> <!-- number to dial, including prefix -->
    <speed>5</speed> <!-- 0=110bps, 2=300bps, 4=1200bps -->
    <!-- 5=2400bps, 6=4800bps, 7=9600bps -->
    <parity>0</parity> <!-- 0=none, 1=odd, 2=even, 3=mark -->
    <!-- 4=space -->
    <databits>3</databits> <!-- 2=7bits, 3=8bits -->
    <stopbits>1</stopbits>
    <flowcontrol>0</flowcontrol> <!-- 0=none, 1=xonxoff, 2=ctsrts -->
    <!-- 3=dsrdtr -->
  </paging-central>
</alpha-pager>

```

3.9 SSL Support on Web Application

Since the Helix Web Application is pure HTML based, the GUI component can be accessed using both regular and secure (SSL) HTTP protocol. Use the following steps to setup SSL support in Helix (replace HELIX_HOME with the correct installation directory name):

1. The application server (Jakarta Tomcat) used by Helix uses a JKS format keystore. Helix by default ships with a keystore with self-signed certificate. If you are not ready to install a valid key yet, you can skip to step 9. Otherwise, first rename or move the existing keystore under

```
HELIX_HOME/etc/helix.keystore
```

2. Create a private/public (RSA) key pair using the following command:

```
HELIX_HOME/jdk/bin/keytool -genkey -keyalg RSA -storepass  
changeit -alias tomcat -keystore HELIX_HOME/etc/helix.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:

```
HELIX_HOME/jdk/bin/keytool -certreq -storepass changeit -alias  
tomcat -keystore HELIX_HOME/etc/helix.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
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:

```
HELIX_HOME/jdk/bin/keytool -import -v -trustcacerts -alias
tomcat -storepass changeit -file my_new_cert.pem -keystore
HELIX_HOME/etc/helix.keystore
```

8. When asked "Trust this certificate?", answer yes and The certificate will be installed into the keystore.

Verify that the certificate has been imported correctly using:

```
HELIX_HOME/jdk/bin/keytool -list -v -storepass changeit -
keystore HELIX_HOME/etc/helix.keystore
```

9. Edit `HELIX_HOME/tomcat/conf/server.xml` using a text editor (eg. vi, wordpad) and locate the following section (for Helix version 3.4.2 or earlier):

```
<Connector
className="org.apache.tomcat.service.PoolTcpConnector">
<Parameter name="handler"
value="org.apache.tomcat.service.http.HttpConnectionHandler"/>
<Parameter name="port" value="443"/>
<Parameter name="socketFactory"
value="org.apache.tomcat.net.SSLSocketFactory" />
<Parameter name="keystore"
value="HELIX_HOME/etc/helix.keystore" />
<Parameter name="keypass" value="changeit"/>
</Connector>
```

On Helix 3.6 or later, the following section should be located:

```
<Connector
className="org.apache.coyote.tomcat4.CoyoteConnector"
port="443" minProcessors="20" maxProcessors="80"
enableLookups="false" acceptCount="100" debug="0"
scheme="https" secure="true" useURISValidationHack="false"
disableUploadTimeout="true">
<Factory
className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
clientAuth="false" protocol="TLS" keystorePass="chageit"
keystoreFile="/usr/local/helix/etc/helix.keystore"/>
</Connector>
```

which should be commented out by default. Remove the comment (`<!-- . . -->`) and make sure that the keystore, keypass and port parameters are set correctly

10. Save the file and restart the Web Application (if already running). On Linux/Solaris platform this is accomplished using `HELIX_HOME/etc./webapp.init restart`. On Windows platform use Start -> Programs -> Fidelia Helix -> Individual Components -> Stop/Start Web Application
11. Wait 15-30 seconds for the Web Application to initialize and use your web browser to connect to `https://your_helix_host/` and you should see the Helix login page

3.10 Maintenance Tasks

3.10.1 Database backup/restoration

NOTE: *This section applies to Linux and Solaris platforms only.*

The provisioning server utilizes Poet ObjectServer database while DGE components use MySQL, Oracle, etc. These databases need to be backed up periodically as a safety measure, as it will allow you to fall back to the last backed up version in the event of a database corruption. In normal operating mode, the Poet database may have objects in memory and writing data to the database files randomly. It is not recommended that the database files be backed up while Poet is writing to the databases. Helix comes with a script called

`HELIX_HOME/utils/tasks/01d_50_db_backup.sh` that should be used to backup the database in proper manner. The script 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. Once the backup operation has been completed, the script sends signal to Poet to resume normal operation. While the backup operation is in progress, Poet will continue to operate normally and cache all write transactions.

`HELIX_HOME/utils/tasks/01d_50_db_backup.sh` should be run from the helix cron job

(`HELIX_HOME/utils/runPeriodicTasks.pl`) nightly. (For additional information about Helix cron jobs, see “Scheduled tasks (cron jobs)” on page 16.) By default this script will create a

tar-gzipped archive in the `/var/backup` directory with names of the form `helix-mm-dd-yy,hh-mm.tar.gz`. If you would like to create these files somewhere else, edit the `01d_50_db_backup.sh` script to specify the destination by changing the `backupPath` variable. (Whether you use the default or some other location, make sure that there is sufficient disk space for the backup files.)

□ To manually backup the provisioning database:

```
% cd HELIX_HOME
utils/tasks/01d_50_db_backup.sh
```

In order to restore a copy of the provisioning database, simply uncompress and un-tar the archive into the `HELIX_HOME` directory. Make sure what you do not have the application running while restoring a database.

□ To restore a copy of the provisioning database:

```
% /etc/init.d/helix stop
% cd HELIX_HOME
% cd database
% mv provisioning provisioning.OLD
% cd ..
% gunzip -c /var/backup/helix-mm-dd-yy,hh-mm.tar.gz | tar xvf
- database
% /etc/init.d/helix start
```

□ Backing up the DGE database:

```
bin/mysqldump --defaults-file=HELIX_HOME/etc/mysql.conf --opt
aggregateddatadb > /var/backup/backup-db.sql
```

□ To restore the DGE database:

```
% /etc/init.d/helix.init stop
% cd /tmp
% gunzip -c /var/backup/helix-mm-dd-yy,hh-mm.tar.gz | tar xvf
- aggregateddatadb.sql
```

```

% cd HELIX_HOME
% cd database
% mv aggregateddatadb aggregateddatadb.OLD
% cd ../mysql/bin
% mysql --defaults-file=../../etc/mysql.conf aggregateddatadb
< /tmp/aggregateddatadb.sql
% rm /tmp/aggregateddatadb.sql
% /etc/init.d/helix.init start
  
```

3.11 Operating Helix Behind Firewalls

If Helix 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).

3.11.1 Requirements for Web Application

The web application is accessed over port 80 or 443:

Table 3.5 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

3.11.2 Requirements for DGE (monitors)

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:

Table 3.6 Firewall rules for a DGE that is behind a firewall

protocol	direction	local port	remote host	remote port	reason
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 E-mail
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

Table 3.6 Firewall rules for a DGE that is behind a firewall

protocol	direction	local port	remote host	remote port	reason
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

3.12 Helix Operation 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.

Helix 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.