

IBM® Tivoli® Netcool/OMNIbus Gateway for
HP ServiceManager
8.0

Reference Guide
November 8, 2013



Notice

Before using this information and the product it supports, read the information in [Appendix A, “Notices and Trademarks,”](#) on page 31.

Edition notice

This edition (SC14-7634-03) applies to version 8.0 of IBM Tivoli Netcool/OMNIbus Gateway for HP ServiceManager and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC14-7634-02.

© **Copyright International Business Machines Corporation 2006, 2013.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

- About this guide..... V**
 - Document control page..... v
 - Conventions used in this guide..... vi

- Chapter 1. Gateway for HP ServiceManager 1**
 - Summary..... 1
 - Supported versions and platforms..... 2
 - Gateway operation..... 2
 - Netcool/OMNIbus Gateway for HP ServiceManager integration architecture..... 3
 - Reader, writer, and route..... 3
 - OS-SC and SC-OS routes..... 3
 - R/W modules..... 4
 - R/W module log files..... 4
 - Feedback field..... 4
 - HP ServiceManager users..... 4
 - Installing the gateway..... 5
 - Installing probes and gateways on Tivoli Netcool/OMNIbus V8.1..... 5
 - Installing the gateway on Tivoli Netcool/OMNIbus V7.4.0..... 6
 - Configuration..... 8
 - AES encryption..... 8
 - Gateway configuration..... 10
 - Create the gateway configuration file..... 11
 - Create a gateway conversion table..... 19
 - Edit the gateway action SQL files..... 20
 - Configure the gateway to connect to ObjectServers running in a failover pair..... 22
 - HP OpenView Connect-It (Integration Server)..... 22
 - Connection scenarios..... 22
 - Event format..... 23
 - Event queues..... 23
 - Event services..... 23
 - Incident management system..... 23
 - HP ServiceManager record category..... 24
 - Bidirectional implementation..... 24
 - Running the gateway..... 24
 - Running the gateway on UNIX..... 24
 - Error messages..... 24

- Appendix A. Notices and Trademarks..... 31**
 - Notices..... 31
 - Trademarks..... 32

About this guide

The following sections contain important information about using this guide.

Document control page

Use this information to track changes between versions of this guide.

The IBM Tivoli Netcool/OMNIBus Gateway for HP ServiceManager documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM Tivoli Netcool/OMNIBus Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSHTQ/omnibus/common/kc_welcome-444.html?lang=en

Document version	Publication date	Comments
SC23-6399-01	July 9, 2007	First IBM publication.
SC23-6399-02	August 10, 2007	Third party screen shots removed from the document. Descriptions of both reader and writer scenarios updated for additional clarity.
SC23-7942-01	April 11, 2008	“Summary” on page 1 section updated. STORE_FILE write attribute updated. Figure1: HP OpenView ServiceCenter Architecture updated. Example configuration file updated. CURR_SYS_SEQ_ TABLE and CURR_SYS_SEQ_ NAME attributes removed. Configuring the gateway for failover processing section removed.

Table 1. Document modification history (continued)		
Document version	Publication date	Comments
SC14-7634-00	October 29, 2010	Sybase support information removed ServiceManager Information added throughout the document “ Summary ” on page 1 updated “ Supported versions and platforms ” on page 2 updated “ Write attributes ” on page 11 updated “ Connection scenarios ” on page 22 updated
SC14-7634-01	January 20, 2012	Information about using this guide added in “ Conventions used in this guide ” on page vi. “ Connection scenarios ” on page 22 updated.
SC14-7634-02	November 30, 2012	Guide updated for Netcool/OMNIbus V7.4 release. “ Installing the gateway ” on page 5 updated.
SC14-7634-03	November 8, 2013	“ Summary ” on page 1 updated.

Conventions used in this guide

All gateway guides use standard conventions for operating system-dependent environment variables and directory paths.

Operating system-dependent variables and paths

All gateway guides use standard conventions for specifying environment variables and describing directory paths, depending on what operating systems the gateway is supported on.

For gateways supported on UNIX and Linux operating systems, gateway guides use the standard UNIX conventions such as `$variable` for environment variables and forward slashes (/) in directory paths. For example:

```
$OMNIHOME/gates
```

For gateways supported only on Windows operating systems, gateway guides use the standard Windows conventions such as `%variable%` for environment variables and backward slashes (\) in directory paths. For example:

```
%OMNIHOME%\gates
```

For gateways supported on UNIX, Linux, and Windows operating systems, gateway guides use the standard UNIX conventions for specifying environment variables and describing directory paths. When using the Windows command line with these gateways, replace the UNIX conventions used in the guide

with Windows conventions. If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Note : The names of environment variables are not always the same in Windows and UNIX environments. For example, %TEMP% in Windows environments is equivalent to \$TMPDIR in UNIX and Linux environments.

Operating system-specific directory names

Where Tivoli Netcool/OMNIbus files are identified as located within an *arch* directory under NCHOME or OMNIHOME, *arch* is a variable that represents your operating system directory. For example:

`$OMNIHOME/platform/arch`

The following table lists the directory names used for each operating system.

Note : This gateway may not support all of the operating systems specified in the table.

Operating system	Directory name represented by <i>arch</i>
AIX® systems	aix5
Red Hat Linux® and SUSE systems	linux2x86
Linux for System z®	linux2s390
Solaris systems	solaris2
Windows systems	win32

OMNIHOME location

Gateways and older versions of Tivoli Netcool/OMNIbus use the OMNIHOME environment variable in many configuration files. Set the value of OMNIHOME as follows:

- On UNIX and Linux, set \$OMNIHOME to \$NCHOME/omnibus.
- On Windows, set %OMNIHOME% to %NCHOME%\omnibus.

Chapter 1. Gateway for HP ServiceManager

The Gateway for HP ServiceManager is a fully functional bidirectional gateway.

Alerts forwarded from the ObjectServer through the gateway form HP ServiceManager Incident Management tickets. Both systems work together to create and update alerts and tickets. Any change made to a alert or a HP ServiceManager Incident Management ticket is reflected in its associated alert or ticket.

This guide contains the following sections:

- [“Summary” on page 1](#)
- [“Gateway operation” on page 2](#)
- [“HP ServiceManager users” on page 4](#)
- [“Installing the gateway” on page 5](#)
- [“Gateway configuration” on page 10](#)
- [“Running the gateway” on page 24](#)
- [“Error messages” on page 24](#)

Summary

The following table provides a summary of the Gateway for HP ServiceManager.

Gateway target 8.0	HP ServiceManager 9.3 running on HP Connect-It 9.4 and earlier versions of both software. Note : The gateway has not been tested on later versions of either software.
Gateway executable name	nco_g_peregrine nco_cirwmodule
Patch number	8.0
Gateway supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: http://www-01.ibm.com/support/docview.wss?uid=swg21674676
Configuration files	\$OMNIHOME/gates/peregrine/ G_PEREGRINE.conf \$OMNIHOME/gates/peregrine/ open_action.peregrine.sql \$OMNIHOME/gates/peregrine/ close_action.peregrine.sql \$OMNIHOME/gates/peregrine/ update_action.peregrine.sql \$OMNIHOME/gates/peregrine/sys_seq.sql

<i>Table 3. Summary (continued)</i>	
Requirements	A currently support version of Tivoli Netcool/OMNIbus
Internationalization	Available (Solaris only)

Supported versions and platforms

The following table shows the supported Gateway for HP ServiceManager versions and the platforms on which they run.

<i>Table 4. Supported versions and platforms</i>		
Gateway version	BinariesPlatform	Platform
HP ServiceManager 9.3 running on HP Connect-It 9.4 and earlier versions of both software.	nco_g_peregrine	Solaris 9, 10 AIX 5.3, 6.1 HP-UX 11i Linux 4.0 (of Red Hat) Linux 9.0, 10.0 (of SLES)

Gateway operation

This section describes the operation of the IBM Tivoli Netcool/OMNIbus Gateway for HP ServiceManager.

The gateway manages three types of events, each of which performs a different ticket function. These are:

- Open (PMO)
- Update (PMU)
- Close (PMC)

The PMO event type creates a HP ServiceManager ticket and forwards it to the HP ServiceManager Incident Management System. The Incident Management System issues a response to the ObjectServer to confirm that the ticket has been created.

The PMU event type updates a ticket if a change is made to an associated event in the ObjectServer. This ensures that tickets and alerts are synchronized.

The PMC event type closes a ticket if the associated event is resolved.

When a ticket is created using PMO, the gateway stores the serial number of the alert. When the Incident Management system responds, indicating that the ticket has been created, the gateway stores the number of the ticket.

By storing the serial number of the alert and the Incident Management Problem ID together, the gateway can track associated alerts and tickets.

Netcool/OMNIbus Gateway for HP ServiceManager integration architecture

The IBM Tivoli Netcool/OMNIbus Gateway for HP ServiceManager and HP ServiceManager are integrated in the following way.

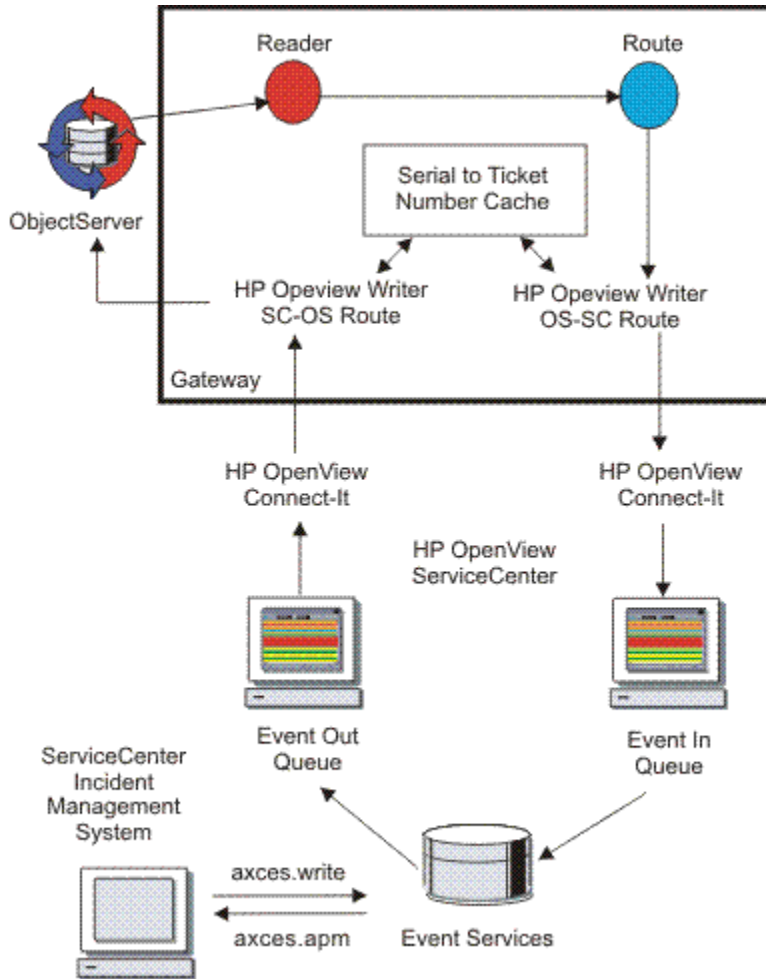


Figure 1. HP OpenView ServiceCenter architecture

Reader, writer, and route

The Gateway for HP ServiceManager uses a standard reader and route which are defined in the gateway configuration file.

The writer defines four different mappings, one for each event type (PMO, PMU, PMC), and one for HP ServiceManager journal entries. You need to create a gateway configuration file that contains these components. Refer to [“Create the gateway configuration file”](#) on page 11 for more information about how to do this.

OS-SC and SC-OS routes

The ObjectServer to ServiceCenter (OS-SC) and ServiceCenter to ObjectServer (SC-OS) routes define the interaction between the gateway and ServiceCenter.

They work in conjunction to track tickets and alerts. To do this, they use the serial to Incident Management Problem ID cache to match alerts routed from the ObjectServer with tickets routed from HP ServiceManager .

For the gateway to map events to HP ServiceManager correctly, you need to define field conversions between the gateway and HP ServiceManager. Refer to [“Create a gateway conversion table”](#) on page 19 for more information about how to do this.

HP ServiceManager can feed back events through the gateway into the ObjectServer. You need to define how the gateway should act when it receives an event using SQL. Refer to [“Edit the gateway action SQL files” on page 20](#) for more information about how to do this.

R/W modules

The OS-SC and SC-OS routes also launch Read/Write modules (outside the gateway) which read and write tickets and alerts to and from HP ServiceManager.

The OS-SC and SC-OS routes also launch Read/Write modules by creating TCP/IP connections to HP OpenView Connect-It. Both of these modules produce a log file to record the information transferred through the gateway, as described in [“R/W module log files” on page 4](#).

R/W module log files

The R/W modules both produce a log file which records the operation of the gateway.

The following table shows the fields that are sent to the log files:

Field name	Description
evtype	Event type being performed on this record. Can be pmo, pmu, or pmc.
evtime	Event time set and controlled by the HP ServiceManager event manager.
evsysseq	Internal sequence number maintained by the gateway.
evusrseq	User sequence number used by the gateway to determine whether the event was generated by HP ServiceManager, or is a return event from the gateway.
evuser	The username used to read or write Netcool/OMNIbus Gateway for HP ServiceManager and ServiceCenter records.
evpswd	Application password for the event, if one is required.
evsepchar	Character used to separate HP ServiceManager record fields.
evfields	Complete HP ServiceManager record. Fields are delimited by the separator character.

Feedback field

When the gateway sends PMOs to HP ServiceManager, HP ServiceManager responds with the trouble ticket number. However, this is not an immediate action; there is a short delay before the gateway receives the trouble ticket number.

When the gateway sends a PMO, it immediately updates the ObjectServer feedback field (specified by the FEEDBACK_FIELD writer attribute) with a message (specified by the PMO_SENT writer attribute) to indicate that the PMO has been sent and the gateway is waiting for a trouble ticket number.

If the feedback field value for an alert in the ObjectServer is updated to reinsert, the update will be forwarded to HP ServiceManager as a PMO rather than a PMU. This functionality is not required in normal circumstances.

HP ServiceManager users

To configure HP ServiceManager, you must have administration privileges.

Before you run the gateway, you should create an operator named Omnibus. This operator is used in the configuration and allows you to track events that are generated by the gateway.

Installing the gateway

There are separate procedures for installing the gateway on each version of Tivoli Netcool/OMNIBus.

Follow the procedure for the version of Tivoli Netcool/OMNIBus that your site uses.

Installing probes and gateways on Tivoli Netcool/OMNIBus V8.1

From Tivoli Netcool/OMNIBus V8.1 onwards, Tivoli Netcool/OMNIBus probes and gateways can be installed using the IBM Installation Manager. One of the key features of Installation Manager is that all platforms are shipped in a single ZIP file, which means that you do not have to select the platform that you require; Installation Manager does it for you.

Before you can install a probe or gateway, you must have installed and configured Installation Manager and Tivoli Netcool/OMNIBus. To install probes and gateways, you must make sure that the Core Tivoli Netcool/OMNIBus features **Probe Support** and **Gateway Support** respectively are installed.

Installing probes and gateways using the Command Line Tool

To install the probe or gateway using the Command Line Tool, run the following command:

```
installation_manager_location/eclipse/tools/imcl -c install  
com.ibm.tivoli.omnibus.integrations.integration_name -repositories  
repository_containing_required_integration -installationDirectory  
location_of_netcool_omnibus_install_you_are_installing_into
```

Where *integration_name* specifies the name of the probe or gateway that you want to install.

You will be prompted to agree to the terms and conditions of the license as a prerequisite for installing the integration. If you have already reviewed the license and want to skip the manual acceptance, add the `-acceptLicense` option to the `install` command to silently agree to the license.

The following is an example command used to install the SNMP Probe:

```
imcl -c install com.ibm.tivoli.omnibus.integrations.nco-p-mttrapd -  
repositories /home/my_home_dir/nco-p-mttrapd_im_package -  
installationDirecory /opt/IBM/tivoli/netcool
```

Where `/home/my_home_dir/nco-p-mttrapd_im_package` contains the unzipped contents of the SNMP Probe Installation Manager package.

Note : The command line tool does not add the repository permanently to the Installation Manager instance. If you subsequently start the Installation Manager GUI, the repositories will not be present in the **Repositories** dialog box.

Uninstalling probes and gateways using the Command Line Tool

To uninstall the probe or gateway using the Command Line Tool, run the following command:

```
installation_manager_location/eclipse/tools/imcl uninstall  
com.ibm.tivoli.omnibus.integrations.integration_name -installationDirectory  
location_of_netcool_omnibus_install_you_are_uninstalling_from
```

Where *integration_name* specifies the name of the probe or gateway that you want to uninstall.

The following is an example command used to uninstall the SNMP Probe:

```
imcl uninstall com.ibm.tivoli.omnibus.integrations.nco-p-mttrapd -  
installationDirecory /opt/IBM/tivoli/netcool
```

Installing probes and gateways using the GUI

To install the probe or gateway using the GUI, use the following steps:

1. Unzip the IM package that contains the probe or gateway into a directory of your choosing. A file called `repository.config` will appear after unzipping the IM package.
2. Start Installation Manager using the following command:

```
installer_path/IBMIM
```

Where *installer_path* is the path to the Installation Manager directory.
3. Perform the following menu actions to display the repository dialog box:
Files > Preferences > Repositories.
4. Use the button **Add Repository** in the repository dialog box to point to the repository that contains the unzipped IM package that contains the probe or gateway. This is the repository that contains the `repository.config` file.
5. Click the **Install software packages** icon.
6. Select the name of the probe or gateway that you want to install.
7. Click **Next**.
8. Click **I accept** when the Licensing panel appears.
9. Highlight **IBM Tivoli Netcool OMNIBus** in the **Package Group Name** field.
10. Click **Next**.
11. Click **Next**.
12. Click **Install**.
13. When the **Install Packages** panel appears indicating that you have successfully installed the probe or gateway, click **Finish**.

Uninstalling probes and gateways using the GUI

To uninstall the probe or gateway, use the following steps:

1. Start Installation Manager using the following command:

```
installer_path/IBMIM
```

Where *installer_path* is the path to the Installation Manager directory.
2. Click the **Uninstall software packages** icon.
3. Select the name of the probe or gateway that you want to uninstall.
4. Click **Next**.
5. Click **Uninstall**.
6. When the **Install Packages** panel appears indicating that you have successfully uninstalled the probe or gateway, click **Finish**.

Installing the gateway on Tivoli Netcool/OMNIBus V7.4.0

For Tivoli Netcool/OMNIBus V7.4.0, all gateways are installed using the Tivoli Netcool/OMNIBus installer.

You can install the gateway using any of the following:

- [“The installation wizard” on page 7](#)
- A text-based installer ([“Console mode” on page 7](#))
- Settings predefined in a text file ([“Silent mode” on page 7](#))

The installation package and patches for the gateway are supplied as archives. The archive management application that you use to extract the files must be able to preserve the directory structure contained in the archive on extraction.

Note : If you are installing a 32-bit gateway on a system that runs a 64-bit UNIX or Linux operating system, you will need to install additional, 32-bit operating system libraries. See the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide* for more information.

Obtaining the installation package

To obtain the installation package and prepare it for installation use the following steps:

1. Download the installation package for the gateway from the Passport Advantage Online Web site:
http://www-306.ibm.com/software/howtobuy/passportadvantage/pao_customers.htm
2. Make a backup of any existing configuration files that you want to retain.
3. Extract the contents of the installation package to a temporary directory.

Now use one of the installation methods to install your gateway. In each case, the gateway is installed in the following directory:

```
$NCHOME/omnibus/gates
```

The installation wizard

To install the gateway using the installation wizard:

1. Run the installer for your operating system:

```
$NCHOME/omnibus/install/nco_install_integration
```
2. When the installation wizard starts, specify the extracted directory that contains the README .txt file as the location of the gateway installation files.
3. Accept the license conditions.

Console mode

To install the gateway in console mode:

1. Run the installer for your operating system:

```
$NCHOME/omnibus/install/nco_install_integration -i console
```
2. When the text-based installer starts, specify the extracted directory that contains the README .txt file as the location of the gateway installation files.
3. Accept the license conditions.

Silent mode

To install the gateway in silent mode:

1. Create a text file named `response.txt` and add the following entries:

```
PROBE_OR_GATE_LOCATION=README_directorypath  
LICENSE_ACCEPTED=true
```

where *README_directorypath* is the path to the directory containing the README .txt file in the extracted package.

2. Run the installer for your operating system:

```
$NCHOME/omnibus/install/nco_install_integration -i silent -f  
response_path/response.txt
```

where *response_path* is the full path to the `response.txt` file.

Configuration

After installing the gateway, perform the following steps.

1. Create the server G_PEREGRINE in the interfaces file using `nco_xigen`.

Note : For details about using `nco_xigen`, see the *IBM Tivoli Netcool/OMNIbus Administration Guide*,

2. Copy the configuration file (`G_PEREGRINE.conf`) to the following directory: `$OMNIHOME/etc`.
3. Edit the configuration file to suit your environment.

AES encryption

AES encryption can be used to encrypt any string within the gateway writer section of the configuration file. It is used by the gateway to prevent sensitive data from being available in readable format in the gateway configuration file.

Note : AES encryption is supported on Tivoli® Netcool/OMNIbus V7.4.0 (and above) on all operating systems.

`nco_aes_crypt`

You can encrypt strings in the gateway configuration file using the `nco_aes_crypt` tool (supplied with Tivoli Netcool/OMNIbus). The syntax of encrypted data is as follows:

```
@datalength:encrypted_data@
```

Where *datalength* is the length of the data in bytes (expressed as a decimal) and the data itself is base64 encoded. The at sign (@) indicates the start and end of the encrypted data definition. The colon (:) acts as a field separator.

The encrypted values appear in single quotes on the right side of expressions in the writer section of the configuration file. The following is an example line from a configuration file showing the host name given in encrypted format:

```
HOST = '@64:1HBLuIPLNye8zCWhykFVFY7y90V9kCjGK5GSWu5VBdSlgQ0qarq6T4UK4xk5Vqix@'
```

Note : You can obtain the `nco_aes_crypt` tool from the IBM Passport Advantage website: http://www-306.ibm.com/software/howtobuy/passportadvantage/pao_customers.htm. Access the Software Downloads section and search for Netcool/OMNIbus Gateway configuration encryption library.

Using the `nco_aes_crypt` tool

Data in the configuration file must be encrypted using the `nco_aes_crypt` tool.

This is a command line tool which takes the following format:

```
nco_aes_crypt [-d | -e] [-o outfile] -k keyfile -f filename  
nco_aes_crypt [-d | -e] [-o outfile] -k keyfile data
```

The output of this command will be the encrypted string to be used in the configuration file.

The following table describes the options available with `nco_aes_crypt`:

Command line option	Description
-d or -e	Use this option to specify the mode in which the nco_aes_crypt tool runs: d - decrypt mode e - encrypt mode The default is e.
-o <i>string</i>	Use this option to specify the output file to which the encrypted data will be written.
-k <i>string</i>	Use this option to specify the path of the file containing the key data.
-f <i>string</i>	Use this option to specify the path of the file containing data requiring encryption.
<i>data</i>	Use this option to specify the data to be encrypted or decrypted.

Encryption key file

The encryption key is stored in a flat file alongside the encrypted data. The key storage file has an ASCII numeric key length indicator followed by a colon and the key in binary form.

The format of the key file is as follows:

```
key_length:key_data
```

Where `key_length` is the length of the key in bits and the `key_data` is the key in binary form. Valid length values are 128, 192 and 256.

For example:

```
128:1234567812345678
```

In this case, `key_length` is 128 since the ASCII string `1234567812345678` has 16 bytes (128 bits).

You can generate random or pre-defined keys of varying lengths using `nco_keygen`. To generate a key file, use the following command:

```
nco_keygen -o outfile[-l length|-k]key[-h |-?]
```

The following table gives the descriptions of the above command line options.

Command line option	Description
-o <i>outfile</i>	Use this option to specify the output file name.
-l <i>length</i>	Use this option to specify the length (in bits) of the key to write out. The default is 128. Note : The value that you specify must be divisible by 8.

Table 7. Encryption key file command line options (continued)	
Command line option	Description
-k key	Use this option to specify the key to be written out, expressed as hex digits. Note : This option bypasses automatic key generation.
-h /-?	Use this option to print the help information and exit

Note : AES encryption is used as the initial encryption method for sensitive data. However, this does not mean that the data can be considered to be secure purely due to AES encryption; the security of the data depends on the restriction of access to the key file used for AES encryption. Access to this file is controlled using UNIX file permissions.

Using encrypted data

To use encrypted data, you must add the following line to the WRITER section of the configuration file:

```
LOAD ENCRYPTION KEY FROM 'key_file_path' USING 'AES'
```

Where *key_file_path* is the path to the file containing the encryption key.

Running the ObjectServer in a secure mode

When the gateway connects to the ObjectServer running in secure mode, it needs to authenticate with a user name and password. This user name and password can be encrypted using the `nco_aes_crypt` tool.

To enable the encryption, the location of the key file must be specified at the beginning of the configuration file. This is followed by the `AUTH_USER` and `AUTH_PASSWORD` fields which contain the encrypted user name and password required for authentication.

The following example shows the three fields that need to be added at the beginning of the configuration file when the ObjectServer runs in a secure mode:

```
LOAD ENCRYPTION KEY FROM '/HOME/72/solaris/omnibus/keyfile_name' USING
'AES';
AUTH_USER '@44:2yXgd6fp9q1Ey4sSAb2RibzA3+PpCZmhAZXo6nNdkvQ=@'; #
encrypted_user_name
AUTH_PASSWORD '@44:mdyEb8VTh+2wALnN1R7dnGnxRZ3BkMQbR5IgxL1Huc=@'; #
encrypted_password
```

Gateway configuration

The `lis` involved in configuring the gateway are discussed in this section.

Before you can run the gateway, you need to configure the gateway and HP ServiceManager.

To do so, use the following `lis`:

1. [“Create the gateway configuration file” on page 11](#)
2. [“Create a gateway conversion table” on page 19](#)
3. [“Edit the gateway action SQL files” on page 20](#)
4. [“Configure the gateway to connect to ObjectServers running in a failover pair” on page 22](#)

Create the gateway configuration file

The Gateway for HP ServiceManager configuration file contains four different mappings, one for each type of event or ticket and one to update HP ServiceManager journals.

The gateway writer attributes and mappings are described in the following sections:

- [“Write attributes” on page 11](#)
- [“Example configuration file” on page 15](#)
- [“Create Open Map example” on page 17](#)
- [“Create Update Map example” on page 18](#)
- [“Create Close Map example” on page 18](#)
- [“Create Journal map” on page 19](#)

Write attributes

The following table describes the attributes used to create a HP OpenView writer:

Attribute name	Type	Description
CLOSE_ACTION_SQL	<i>string</i>	Name of the file containing the close action SQL to be performed at the ObjectServer. No action is taken if no attribute is supplied. Used in bidirectional implementations of this gateway.
CLOSE_MAP	<i>string</i>	Name of the close map.
CLOSE_PROBLEMS	<i>boolean</i>	Specifies whether or not the Netcool/OMNIbus Gateway for HP ServiceManager can close problem records in HP ServiceManager . TRUE enables problem closure. FALSE disables problem closure.
CONVERSIONS_ TABLE	<i>string</i>	Name of the conversions table in the Netcool/OMNIbus Gateway for HP ServiceManager ObjectServer that holds the data values of required conversions between HP ServiceManager and Netcool/OMNIbus Gateway for HP ServiceManager fields.
COUNTERPART	<i>string</i>	Name of the counterpart reader in a bidirectional gateway configuration. This defines a link between the reader and the writer of one half of the bidirectional gateway and prevents the gateway from sending an alert straight back to the ObjectServer from which it was received. There is no default value for this attribute. Note : The reader must be running for the attribute value to be applied.

Table 8. Writer attributes (continued)

Attribute name	Type	Description
DATE_FORMAT	<i>string</i>	Format of date required. The default is %B %d, %Y %I:%M:%S%p. Note : For details of this format are available by looking at the manual page for strftime C call.
FEEDBACK_FIELD	<i>string</i>	Name of the field in the feedback ObjectServer where the HP ServiceManager Incident Management problem ID should be written.
FEEDBACK_SERVER	<i>string</i>	Name of the ObjectServer where the gateway feedback mechanism writes the HP ServiceManager problem record.
FILTER_CONDITION	<i>string</i>	Enables you to qualify which records are retrieved when the cache is being rebuilt on startup. The default is "" . Note : The gateway ignores this attribute if the REBUILD_CACHE attribute is set to FALSE.
FORWARD_DELETES	<i>boolean</i>	Indicates whether the gateway forwards delete events to HP ServiceManager . The default is TRUE. Note : If this attribute is set to FALSE, the gateway drops delete events.
FORWARD_HISTORIC _ JOURNAL	<i>boolean</i>	Specifies whether or not the gateway forwards all historic journals relating to an alert on first receipt of its problem number from HP ServiceManager . The default is FALSE.
FORWARD_INSERTS	<i>boolean</i>	Indicates whether the gateway forwards insert events to HP ServiceManager . The default is TRUE. Note : If this attribute is set to FALSE, the gateway drops insert events.
FORWARD_JOURNALS	<i>boolean</i>	Indicates whether the gateway forwards journal events to HP ServiceManager . The default is TRUE. Note : If this attribute is set to FALSE, the gateway drops journal events.
FORWARD_UPDATES	<i>boolean</i>	Indicates whether the gateway forwards update events to HP ServiceManager . The default is TRUE. Note : If this attribute is set to FALSE, the gateway drops update events.

Table 8. Writer attributes (continued)

Attribute name	Type	Description
HOPEFUL_PMC_CLOSE	<i>boolean</i>	This turns off the forwarding of the PMC event when the alert deletion details are incomplete. The default is TRUE.
JOURNAL_MAP	<i>string</i>	Name of the journal map.
MAX_SAF_FILE_SIZE	<i>integer</i>	Maximum size (in bytes) of the file that the gateway uses to store events when the connection to HP ServiceManager is lost. If the file reaches this size, the gateway shuts down. The default is 10485760 (10 MB). Note : The minimum allowable value is 1048576 bytes (1MB).
MODULE_REVISION	<i>integer</i>	Allows you to specify the version of the <code>nco_cirwmodule</code> that you are running. The default is 5. Note : To use the most recent functionality added to the gateway, you should leave this attribute set to its default value.
NCO_CIRWMODULE_MAX_BATCH_SIZE	<i>string</i>	Use this property to specify the maximum size of Allows you to specify the version of the <code>nco_cirwmodule</code> that you are running. The default is 100.
OPEN_ACTION_SQL	<i>string</i>	Name of the file containing the open action SQL to be performed at the ObjectServer. No action is taken if no attribute is supplied. Used in bidirectional implementations of this gateway.
OPEN_MAP	<i>string</i>	Name of the open map.
PMC_EVENT_TYPE_ATTR	<i>string</i>	Allows you to specify an alternative name for close events. If you omit this attribute, the close event type defaults to <code>pmc</code> .
PMO_EVENT_TYPE_ATTR	<i>string</i>	Allows you to specify an alternative name for open events. If you omit this attribute, the open event type defaults to <code>pmo</code> .

Table 8. Writer attributes (continued)

Attribute name	Type	Description
PMO_SENT_MESSAGE	<i>string</i>	Text that the gateway writes to the feedback field of the ObjectServer when an alert is forwarded for insert into HP ServiceManager . When HP ServiceManager responds with a problem number, this text is overwritten by the problem number. The default is pmo sent. Note : This attribute should not be omitted.
PMU_EVENT_TYPE_ ATTR	<i>string</i>	Allows you to specify an alternative name for update events. If you omit this attribute, the update event type defaults to pmu.
PROBNUM_FIELD_ INDEX	<i>integer</i>	Field index of the returned HP ServiceManager PMO response event where the problem management problem ID can be found.
REBUILD_CACHE	<i>boolean</i>	Specifies whether the gateway queries the ObjectServer and rebuilds the cache on startup. The default is FALSE.
READER_PORT	<i>integer</i>	Port number on the Connect-It to which the gateway reader listens. This corresponds to the WRITER_PORT option on the Connect-It interface. The default is 9000.
REVISION	<i>integer</i>	Revision of writer. You must set this attribute to 1.
REQUIRE_TT	<i>boolean</i>	Specifies whether the gateway drops the update, journal, and delete entries for events that do not contain a HP OpenView ticket number in their feedback fields. The default is FALSE.
RW_MODULE_PING	<i>boolean</i>	Specifies whether the reader module (nco_cirwmodule) periodically checks the status of the gateway binary (nco_g_peregrine). If the reader modules finds that the gateway binary has stopped, the reader module also stops. The default is TRUE.
SEPARATOR_CHAR	<i>string</i>	Character used to separate fields in a HP ServiceManager event record. The default is ^ character.
TYPE	<i>string</i>	Type of writer. This attribute must be set to PEREGRINE.

Table 8. Writer attributes (continued)

Attribute name	Type	Description
UPDATE_ACTION_SQL	string	Name of the file containing the update action SQL to be performed at the ObjectServer. No action is taken if no attribute is supplied. Used in bidirectional implementations of this gateway.
UPDATE_MAP	string	Name of the update map.
WRITER_PORT	integer	Port number on the Connect-It to which the gateway writes the data. The default is 9001.

Example configuration file

The following example shows a configuration file complete with gateway commands and writer attributes:

```
#
START READER SERVER_READER CONNECT TO NCOMS;

#
# Start up the writer
#
START WRITER PEREGRINE_WRITER
(
  TYPE                = PEREGRINE,
  REVISION             = 1,
  MODULE_REVISION     = 5,
  OPEN_MAP             = PEREGRINE_OPEN_MAP,
  UPDATE_MAP          = PEREGRINE_UPDATE_MAP,
  CLOSE_MAP           = PEREGRINE_CLOSE_MAP,
  JOURNAL_MAP         = PEREGRINE_JOURNAL_MAP,
  READER_PORT         = 9000,
  WRITER_PORT         = 9001,
  FEEDBACK_SERVER     = 'NCOMS',
  FEEDBACK_FIELD      = 'Location',
  PROBNUM_FIELD_INDEX = 2,
  SEPARATOR_CHAR      = '^',
  CLOSE_PROBLEMS      = TRUE,
  DATE_FORMAT         = '%m/%d/%y %T',
  COUNTERPART         = SERVER_READER,
  CONVERSIONS_TABLE   = 'conversions.peregrine',
  MAX_SAF_FILE_SIZE   = 20240000,
  STORE_FILE          = '/tmp/G_PEREGRINE_PEREGRINE_WRITER.store',
  OPEN_ACTION_SQL     = '$OMNIHOME/gates/peregrine
/open_action.peregrine.sql',
  UPDATE_ACTION_SQL   = '$OMNIHOME/gates/peregrine
/update_action.peregrine.sql',
  CLOSE_ACTION_SQL    = '$OMNIHOME/gates/peregrine
/close_action.peregrine.sql',
  # STORE_FILE        = '/opt/Omnibus/var/G_PEREGRINE_PEREGRINE_WRITER.store',
  # OPEN_ACTION_SQL   = '/opt/Omnibus/gates/peregrine/
open_action.peregrine.sql',
  # UPDATE_ACTION_SQL = '/opt/Omnibus/gates/peregrine/
update_action.peregrine.sql',
  # CLOSE_ACTION_SQL  = '/opt/Omnibus/gates/peregrine/
close_action.peregrine.sql',
  # OPEN_ACTION_SQL   = 'C:\\Program Files\\Netcool\\OMNIBus
\\gates\\Peregrine\\open_action.peregrine.sql',
  # UPDATE_ACTION_SQL = 'C:\\Program Files\\Netcool\\OMNIBus
\\gates\\Peregrine\\update_action.peregrine.sql',
  # CLOSE_ACTION_SQL  = 'C:\\Program Files\\Netcool\\OMNIBus
\\gates\\Peregrine\\close_action.peregrine.sql',

  # CURR_SYS_SEQ_TABLE = 'custom.sys_seq',
  # CURR_SYS_SEQ_NAME  = 'peregrine',
  # RW_MODULE_PING     = TRUE
);

#
# Add a route from the reader to the writer so the info gets passed
#
```

```
ADD ROUTE FROM SERVER_READER TO PEREGRINE_WRITER;
```

```
#  
# End of file
```

ServiceCenter writer mapping

Mapping techniques define how the format of ObjectServer and HP ServiceManager data should be changed so that it is written in the correct format and to the correct field in the receiving application.

The IBM Tivoli Netcool/OMNIbus Gateway for HP ServiceManager allows individual mapping configurations to be specified for the following functions:

- open map (for PMO event types)
- update map (for PMU event types)
- close map (for PMC event types)
- journal map (for all event types)

Mapping configurations for use with the HP ServiceManager writer must use the following syntax:

```
CREATE MAPPING mappingname  
(  
  FieldId = '@fieldname' [ CONVERT TO DATE ], ... ,  
  [ FieldId = 'fieldname'], ... ,  
  [ FieldId = variablename], ...  
) ;
```

Where:

- `mappingname` is the name of the map configuration to be created.
- `FieldId` is an integer value assigned to this field. It corresponds to a field position in a HP ServiceManager input map for an event of a given type. For details of mapping field positions, see [“HP ServiceManager mapping field position” on page 17](#).
- `@fieldname` is a string value which must be the name of a field in the ObjectServer `alerts.status` table.
- `fieldname` is the value of a static variable that cannot be changed by either Netcool/OMNIbus HP ServiceManager or ServiceCenter.
- `variablename` is the name of a system variable supplied by the gateway. You can specify the following variables:
 - `PROBLEM_NUMBER`
 - `SERVER_NAME`
 - `SERVER_SERIAL`
 - `JOURNAL_TEXT`

None of these variables can be used in the open map.

Variable values are taken from the gateway cache, except for `JOURNAL_TEXT`.

`PROBLEM_NUMBER` holds the HP ServiceManager incident number for an event. `JOURNAL_TEXT` can only be used in the journal map.

It holds the text of an ObjectServer journal entry. `SERVER_NAME` and `SERVER_SERIAL` hold the ObjectServer, `ServerName`, and `ServerSerial` for an event.

- The optional `CONVERT TO DATE` allows the mapping to define a forced conversion to a date type field.

HP ServiceManager mapping field position

The field position within IBM Tivoli Netcool/OMNIbus Gateway for HP ServiceManager and HP ServiceManager field mappings is important, because the gateway forwards events to HP ServiceManager as a single string.

You must ensure that the gateway mapping fields are in the same order as the ServiceCenter mapping fields. As each record is processed, the corresponding field values are substituted.

For example, the following assignment specifies that the third ServiceCenter field must be substituted with the value of IBM Tivoli Netcool/OMNIbus Gateway for HP ServiceManager field ServerSerial:

```
3= '@ServerSerial'
```

As the mapping is positional, the sequencing of fields is effected by their placement. If you do not wish to use a particular ServiceCenter field, you must supply a separator character (^) to move to the next field. See [“Example configuration file” on page 15](#).

For certain functions, you do not want values to be substituted. For example, when closing a HP ServiceManager record you may only want to substitute certain values and leave the other fields with the value they previously contained. In this case you must assign a null value (' ') to the HP ServiceManager record.

Use of static variables

HP ServiceManager supports the use of static variables. This is a technique for supplying certain values that must not be changed by IBM Tivoli Netcool/OMNIbus Gateway for HP ServiceManager

The following example assignment shows that HP ServiceManager field number ten has a value of network in version 4.0 and above. Earlier versions used equipment.

```
10= 'network',
```

This is a HP ServiceManager record category that should not be changed as it is only applicable to HP ServiceManager . This value remains the same for the lifetime of the record.

The following sections describe mapping techniques for each of the create, update, close, and journal mapping functions.

Create Open Map example

The following mapping configuration is an example of data that is substituted when a record is opened.

In this example, the tenth ServiceCenter field has been assigned the static variable network in version 4.0 and later, which indicates that the gateway uses the network ServiceCenter record category. In earlier versions, it is assigned the static variable equipment. All other fields are created and assigned ObjectServer values.

```
CREATE MAPPING PEREGRINE_OPEN_MAP
(
1= '@Identifier',
2= '@ServerSerial + @ServerName',
3= '@ServerSerial',
4= '@Node',
5= '@Manager',
6= '@Agent',
7= '@AlertGroup',
8= '@AlertKey',
9= '@Severity',
10= 'network',
11= '@StateChange' CONVERT TO DATE,
12= '@FirstOccurrence' CONVERT TO DATE,
13= '@LastOccurrence' CONVERT TO DATE,
14= '@InternalLast' CONVERT TO DATE,
15= '@Poll',
16= '@Type',
17= '@Tally',
18= '@Class',
19= '@Grade',
20= '@Location',
```

```

21= '@OwnerUID',
22= '@OwnerUID',
23= '@Acknowledged',
24= '@NodeAlias'
);

```

Create Update Map example

The following mapping configuration is an example of data that will be substituted when a record is updated.

In this example, the third `ServiceCenter` field has been assigned the gateway variable `PROBLEM_NUMBER`. The tenth `ServiceCenter` field has been assigned the static variable `network` in version 4.0 and later. In earlier versions, it is assigned the static variable `equipment`. All other fields will be substituted with new `ObjectServer` values.

```

CREATE MAPPING PEREGRINE_UPDATE_MAP
(
1= '@Identifier',
2= '@ServerSerial + @ServerName',
3= PROBLEM_NUMBER,
4= '@Node',
5= '@Manager',
6= '@Agent',
7= '@AlertGroup',
8= '@AlertKey',
9= '@Severity',
10= 'network',
11= '@StateChange' CONVERT TO DATE,
12= '@FirstOccurrence' CONVERT TO DATE,
13= '@LastOccurrence' CONVERT TO DATE,
14= '@InternalLast' CONVERT TO DATE,
15= '@Poll',
16= '@Type',
17= '@Tally',
18= '@Class',
19= '@Grade',
20= '@Location',
21= '@OwnerUID',
22= '@OwnerUID',
23= '@Acknowledged',
24= '@Servername',
25= '@ServerSerial'
);

```

Create Close Map example

The following mapping configuration is an example of data that will be substituted when a record is closed.

In this example only the Netcool/OMNIbus Gateway for HP ServiceManager field `@ServerSerial` and `@ServerName` are substituted. `ServiceCenter` field number three has been assigned the gateway variable `PROBLEM_NUMBER`. All other fields have a null value; they will not be substituted with any new values.

```

CREATE MAPPING PEREGRINE_CLOSE_MAP
(
1= '',
2= '@ServerSerial + @ServerName',
3= PROBLEM_NUMBER,
4= '',
5= '',
6= '',
7= '',
8= '',
9= '',
10= '',
11= '',
12= '',
13= '',
14= '',
15= '',
16= ''
);

```

```

17= ' ',
18= ' ',
19= ' ',
20= ' ',
21= ' ',
22= ' ',
23= ' ',
24= ' ',
25= ' ',
22= ' ',
23= ' ',
24= ' ',
);

```

Create Journal map

The journal map facility allows you to maintain a textual history of an event with the IBM Tivoli Netcool/OMNIbus Gateway for HP ServiceManager journal information.

The create journal map command allows you to specify which ObjectServer fields must be populated in which ServiceCenter fields when updating journal information.

```

CREATE MAPPING PEREGRINE_JOURNAL_MAP
(
1= ' ',
2= '@ServerSerial + @ServerName',
3= PROBLEM_NUMBER,
4= ' ',
5= JOURNAL_TEXT,
6= ' ',
7= ' ',
8= ' ',
9= ' ',
10= ' ',
11= ' ',
12= ' ',
13= ' ',
14= ' ',
15= ' ',
16= ' ',
17= ' ',
18= ' ',
19= ' ',
20= ' ',
21= ' ',
22= ' ',
23= ' ',
24= ' ',
25= ' ',
22= ' ',
23= ' ',
24= ' ',
);

```

Create a gateway conversion table

The conversion table facility allows you to specify certain data conversions to take place between particular ServiceCenter and ObjectServer fields.

For example, ServiceCenter users require a status field to be alphabetic and to have a particular value. IBM Tivoli Netcool/OMNIbus Gateway for HP ServiceManager may hold these as numeric values.

All conversions must be supplied before the gateway can work. If you are not using gateway conversions, you must still create a blank conversions file.

Note : The ObjectServer SQL definition file must be updated to include the conversions database. See [“Changes to SQL definition file”](#) on page 20 for further information.

Manually editing the conversion table

The conversion table is ObjectServer SQL based. This means you can make full use of the ObjectServer SQL.

The conversion expects information to be supplied in columnar format, enclosed in parentheses, where the columns indicate:

```
column 1 = key field name
column 2 = column name
column 3 = ObjectServer value
column 4 = conversion value
```

In the following example, the numeric value of the ObjectServer Severity field, either (0, 1, 2, 3, 4, 5) is converted to HP ServiceManager textual values, either (Clear, Indeterminate, Low, Routine, High, Critical). Username values are also being converted. This can be entered by using the SQL input tool, nco_sql.

```
use database conversions;
go
insert into peregrine values ('Severity0', 'Severity', '0', 'Clear');
insert into peregrine values ('Severity1', 'Severity', '1', 'Indeterminate');
insert into peregrine values ('Severity2', 'Severity', '2', 'Low');
insert into peregrine values ('Severity3', 'Severity', '3', 'Routine');
insert into peregrine values ('Severity4', 'Severity', '4', 'High');
insert into peregrine values ('Severity5', 'Severity', '5', 'Critical');
go
insert into peregrine values ('OwnerUID0', 'OwnerUID', '0', 'Root');
insert into peregrine values ('OwnerUID65534', 'OwnerUID', '65534', 'Nobody');
go
```

Changes to SQL definition file

You must make additions to the Gateway for HP ServiceManager SQL definition file to create the conversions database and table.

The following example shows the SQL statements required. Restart the ObjectServer to apply any SQL changes you make.

```
create database conversions;
create table conversions.peregrine
persistent
(
  KeyField char(255),
  Colname char(255),
  OSValue char(255),
  Conversion char(255),
  PRIMARY KEY (KeyField)
);
go
```

Edit the gateway action SQL files

The gateway action SQL files determine what actions should take place at the ObjectServer when there is new or changed HP ServiceManager information.

The files are ObjectServer SQL based; therefore, you have all of the language facilities available to you. You must specify action SQL for each of the event types: open, update, and close.

When the SQL is being processed, data fields are expanded, conversions are resolved, and ObjectServer fields are populated. The following example updates the Severity field:

```
update alerts.status set Severity = $(23:Severity);
```

In the previous statement the data usage is resolved as follows:

1. The \$ symbol signifies an expansion statement and information enclosed within the parentheses must be resolved.

2. The 23 statement specifies that HP ServiceManager field number 23 is being used.
3. The Severity statement specifies that severity conversions must be used and applied. The colon is a statement separator.
4. The ObjectServer field is updated with the resolved value.

Cache fields

The gateway makes extensive use of certain data fields that are kept in cache storage. One of the reasons why this mechanism is provided is to ensure unique identification of Netcool/OMNIbus Gateway for HP ServiceManager alerts at all times. They may or may not be used in HP ServiceManager records. These fields can be used in the SQL at any time and will increase performance.

The cache fields that are always available are:

```
Serial
ServerSerial
ServerName
PROBLEM_NUMBER
```

Their values can be accessed from SQL files using the variables `$(Serial)`, `$(ServerSerial)`, `$(ServerName)`, and `$(PROBLEM_NUMBER)`, respectively.

The date variable `$(DATE_TIME)` can also be used in SQL files. Its value is taken from the `evtime` element of an event forwarded to the gateway from HP ServiceManager. This variable is particularly useful if inserts to `alerts.journal` actions are required in SQL files.

Open action SQL file

The open action SQL file determines what action should be taken at the ObjectServer when a problem ticket is opened.

The following example updates the ObjectServer and sets three data fields. All three fields are resolved, with Severity populated with ServiceCenter field number 23 and applied conversions. The Agent field is equivalent to ServiceCenter field number 9.

```
#
# The HP OpenView Gateway Problem Open Action SQL file
#
update alerts.status set Severity = $(23:Severity), Agent = '$(9)'
where Serial = $(Serial);
```

Update action SQL file

The update action SQL file determines what action should be taken at the ObjectServer when an incident ticket is updated.

The following example updates the ObjectServer and sets two data fields. The Agent field is resolved to a concatenation of ServiceCenter fields numbers 4 and 9.

```
#
# The HP OpenView Gateway Problem Update Action SQL file
#
update alerts.status set Agent = '$(4) $(9)' where Serial = $(Serial);
```

Close action SQL file

The close action SQL file determines what action should be taken at the ObjectServer when an incident ticket is closed.

The following example performs both an ObjectServer delete and update. The delete statement removes the associated event for the incident ticket. The update statement updates any associated events on the same node as ServiceCenter field five with a severity of 1 (clear).

```
#
# The HP OpenView Gateway Problem Close (Deletion) Action SQL file
#
delete from alerts.status where Serial = $(Serial);
update alerts.status set Severity = 1 where Node = '$(5)';
```

Configure the gateway to connect to ObjectServers running in a failover pair

If the gateway connects to two ObjectServers in a failover pair, you need to create a virtual ObjectServer in the interfaces file (`omni.dat`) that references the primary and backup ObjectServers. You then need to specify the virtual ObjectServer in the gateway configuration file (`G_PEREGRINE.conf`). For example, if two ObjectServers (`NCOMS1` and `NCOMS2`) are defined in the interfaces file, add a third entry to the `omni.dat` file for a virtual ObjectServer (`NCOMS3`) as follows:

```
[NCOMS_1]
~
Primary: devhost 4101
~
[NCOMS_2]
~
Primary: devhost 4102
~
[NCOMS_3]
~
Primary: devhost 4101
Backup: devhost 4102
~
```

For details of configuring ObjectServers using the interfaces file, see the IBM® Tivoli Netcool/OMNIBus Administration Guide, (SC23-6371).

HP OpenView Connect-It (Integration Server)

HP OpenView Connect-It is the interface through which the gateway connects to HP ServiceManager using a TCP/IP connection.

The gateway acts as an HTTP Server for both reading from and writing to HP ServiceManager.

The gateway accepts requests from client HP OpenView Connect-It in reader scenarios and writer scenarios. In the context of Connect-It, a scenario is a set of connectors and mappings that describe the host with which Connect-It is communicating and the target system to which data is sent. Scenarios also provide descriptions of the input and output format of the data and details of any data conversions/transformations required.

For full bidirectional event flow, you must configure both a reader scenario and a writer scenario as described in [“Connection scenarios” on page 22](#).

Connection scenarios

To send and receive events two HP Connect-It scenarios must be configured, one for reading events and another for writing events.

For the writer scenario (Gateway --> HP ServiceManager) the delimited text file connector must be configured in write processing mode with the protocol set to HTTP. You can specify the server address on which the gateway is running in the protocol settings of the connector. For example, if the server is named

gateway and the WRITER_PORT attribute in the gateway configuration file is set to 9001, the address would be gateway:9001.

For the reader scenario (HP ServiceManager --> Gateway) the delimited text file connector must be configured in read processing mode with the protocol set to HTTP. You can specify the server address on which the gateway is running in the protocol settings for the connector. For example, if the server is named gateway and the READER_PORT attribute in the gateway configuration file is set to 9000, the address would be gateway:9000.

Note : When configuring gateway reader scenarios with the Service Manager connector, you must check the **Use Schedule Pointer** box otherwise the same event data will be sent repeatedly to the gateway.

Create suitable scenarios for each event type (PMO, PMU or PMC), and replicate the same mappings as found in the mapping sections of the gateway configuration file.

Event format

The gateway sends and receives information in a particular format.

The gateway sends and expects events in the following format

```
^evtype^evsysseq^evuserseq^evfields
```

where:

- ^ is the separator character as configured in the gateway configuration file
- evtype is the HP ServiceManager event type (PMO, PMU or PMC)
- evsysseq is the HP ServiceManager system sequence number
- evuserseq is the HP ServiceManager user sequence number
- evfields are the event fields, separated by the separator character, and following the same ordering as configured in the gateway mapping for each event type.

See the HP OpenView Connect-It documentation for more information.

Event queues

Event In and Event Out queue events (HP ServiceManager entities) are passed to and from HP OpenView Connect-It and Event Services. You can look at the events currently in the queues.

Event services

HP ServiceManager Event Services manages events received from the gateway and sent from the Incident Management System.

You need to register the event types used by the gateway so that HP ServiceManager can interpret and process the events it receives and so that it can feed back tickets to the gateway.

You also need to create mappings so that HP ServiceManager can map events into tickets and tickets into events.

Incident management system

The HP ServiceManager Incident Management System manages tickets and is configured to send back events to the gateway.

You need to configure HP ServiceManager subroutines so that HP ServiceManager can feed back events into the gateway.

HP ServiceManager record category

The Gateway for HP ServiceManager uses the network record category by default.

This is defined in the gateway mapping by the static variable `network` in Gateway for HP ServiceManager versions 4.0 and later. Earlier versions of HP ServiceManager use the equipment record category and static variable.

Bidirectional implementation

In a bidirectional implementation, alerts from IBM Tivoli Netcool/OMNIbus are passed through the gateway to form HP ServiceManager tickets.

Any subsequent changes to those tickets are reflected in the other application. Therefore, changes to alerts in IBM Tivoli Netcool/OMNIbus are reflected in related HP ServiceManager tickets and changes to HP ServiceManager tickets are reflected in related alerts.

Running the gateway

This section describes how to run the Gateway for HP ServiceManager.

Running the gateway on UNIX

Perform the following list to run the Gateway for HP ServiceManager on UNIX.

To run the gateway on UNIX, run the following command:

```
$OMNIHOME/bin/nco_g_peregrine
```

or use the `-config` command line option as follows:

```
$OMNIHOME/bin/nco_g_peregrine -config /path/config_filename
```

Where *path* is the location of the configuration file and *config_filename* is the name of the configuration file.

Error messages

Error messages provide information about problems that occur while running the gateway. You can use the information that they contain to resolve such problems.

The following table contains error message that can be issued by the Gateway for HP ServiceManager. For information about generic error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*, (SC23-6387).

Error	Description	Action
HP OpenView Writer <i>name</i> : Memory allocation failure.	There is insufficient memory available for the writer to perform an operation.	Release memory by stopping unnecessary processes. Increase the machine's available swap space or add more physical memory.

Table 9. Error messages (continued)

Error	Description	Action
HP OpenView Writer <i>name</i> : Failed to re-acquire alert details from OS.	The gateway was unable to acquire the necessary data for its cache when it received a notification for an event not in the cache. This occurs when the gateway is stopped.	Restart the gateway.
HP OpenView Writer <i>name</i> : Invalid data type for problem number feedback field.	The problem number feedback field was found not to be a character field	Specify a feedback field which is a character type.
HP OpenView Writer <i>name</i> : Serial <i>serial_num</i> already in serial Cache. Cannot add.	An insert into the ObjectServer with the same serial occurred. This should not occur; therefore, the record cannot be added to the cache.	Contact IBM Software Support.
HP OpenView Writer <i>name</i> : Serial <i>serial_num</i> not found in serial cache. Cannot Update/Delete.	An event update/delete failed to find the necessary alert details in the writer's serial cache.	Contact IBM Software Support.
HP OpenView Writer <i>name</i> : Failed to construct path to HP OpenView Read/Write Module	Failed to determine the path to the HP OpenView read/write module executable.	Contact IBM Software Support.
HP OpenView Writer <i>name</i> : Failed to find the HP OpenView Read/Write Module name.	Failed to access the HP OpenView read/write module executable in \$OMNIHOME/bin/arch.	Verify the existence and location of module.
HP OpenView Writer <i>name</i> : Incorrect permissions on the HP OpenView module binary name.	The HP OpenView read/write module was found, but does not have execution or read permissions	Change the file with the correct permissions.
HHP OpenView Writer <i>name</i> : Failed to construct the argument list for HP OpenView Module.	Failed to construct the launch request for the HP OpenView read/write module, due to memory limitations.	Release memory by stopping unnecessary processes. Increase the machine's available swap space or add more physical memory.
HP OpenView Writer <i>name</i> : Failed to start the OS-SC Writer.	The ObjectServer to ServiceCenter writer module failed to start.	Examine the writer module log file for further messages.
HP OpenView Writer <i>name</i> : Failed to start the SC-OS Reader.	The ServiceCenter to ObjectServer reader module failed to start.	Examine the reader module log file for further messages.

Table 9. Error messages (continued)

Error	Description	Action
HP OpenView Writer <i>name</i> : Failed to create the Serial Cache Mutex.	Failed to construct the data protection structure due to insufficient resources. This is due to insufficient memory resources.	Release memory by stopping unnecessary processes. Increase the machine's available swap space or add more physical memory.
HP OpenView Writer <i>name</i> : Failed to start the SC-to-OS service thread.	The main HP OpenView writer failed to start the ServiceCenter to ObjectServer handler thread. This is due to insufficient memory resources.	Release memory by stopping unnecessary processes. Increase the machine's available swap space or add more physical memory.
HP OpenView Writer <i>name</i> : Failed to send a SIGTERM to shutdown SC Writer.	Failed to stop the ServiceCenter writer module. This may result in the gateway port being locked.	Find the process in the process list and stop it.
HP OpenView Writer <i>name</i> : Failed to send a SIGTERM to shutdown SC Reader.	Failed to stop the ServiceCenter reader module. This may result in the gateway port being locked.	Find the process in the process list and stop it.
HP OpenView Writer <i>name</i> : Map ' <i>map name</i> ' is not the journal map and cannot contain the 'JOURNAL_TEXT' map item in HP OpenView Writer ' <i>writer name</i> '.	The special mapping name JOURNAL_TEXT was contained within a map that was not specified to be the journal map. The mapping name references the position at which journal text should be positioned in the journal PMU event.	Remove the names from the incorrect map.
HP OpenView Writer <i>name</i> : Failed to send SC Event to the SC Writer module.	A problem was detected on the connection to the ServiceCenter writer module.	Examine the ServiceCenter writer module log file for further messages.
HP OpenView Writer <i>name</i> : Failed to wait for return from the SC Writer module.	The gateway failed to wait for the return message from ServiceCenter.	Examine the ServiceCenter writer module log file to determine the module error.
HP OpenView Writer <i>name</i> : Failed to read the status return message from the SC Writer module.	The gateway failed to read the status return message from ServiceCenter.	Examine the ServiceCenter writer module log file to determine the module error.
HP OpenView Writer <i>name</i> : Failed to send event to ServiceCenter.	The gateway failed to send an event to ServiceCenter.	Examine the ServiceCenter writer module log file to determine the module error.
HP OpenView Writer <i>name</i> : SC Writer Module experienced Fatal Error.	Expected error occurred when the ServiceCenter writer module terminated.	Examine the ServiceCenter writer module log file to determine the module error.

Table 9. Error messages (continued)

Error	Description	Action
HP OpenView Writer <i>name</i> : Failed to build serial index.	The ServiceCenter writer was unable to construct the fast access jump index for the Serial index.	Check the alerts.status table definitions and ensure the field exists.
HP OpenView Writer <i>name</i> : Failed to build server serial index.	The ServiceCenter writer was unable to construct the fast access jump index for the ServerSerial index.	Check the alerts.status table definitions and ensure the field exists.
HP OpenView Writer <i>name</i> : Failed to build server name index.	The ServiceCenter writer was unable to construct the fast access jump index for the ServerName index.	Check the alerts.status table definitions and ensure the field exists.
HP OpenView Writer <i>name</i> : Failed to find field <i>Field Number</i> in SC Event.	The field specified in the action SQL could not be found.	Check the full numbers in the returned event map. Specifying the 25th field in a 24 field list will also generate this error as the 25th field does not exist.
HP OpenView Writer <i>name</i> : Invalid field name for expansion on action SQL.	An unknown field name has been specified in the action SQL for expansion that was neither Serial, ServerSerial, ServerName, or PROBLEM_NUMBER.	Correct action SQL.
HP OpenView Writer <i>name</i> : Unenclosed field expansion request in action SQL action sql.	An unenclosed field expansion was found in the action SQL.	This is an expansion not ended with a close parenthesis. Supply the parenthesis.
HP OpenView Writer <i>name</i> : Failed to send SQL command to ObjectServer. SC-to-OS Feedback failed.	An error occurred when a SQL statement was sent to the feedback ObjectServer.	If the feedback involved the sending of user-defined action SQL, check complete syntax of the action SQL.
HP OpenView Writer <i>name</i> : Failed to correctly extract the problem number for returned problem open event for serial serial number.	The gateway could not successfully extract the problem number for the returned ServiceCenter PMO event.	Contact IBM Software Support.
HP OpenView Writer <i>name</i> : Invalid problem number. Field empty.	The field index specified as the field that will contain the problem number in the returned ServiceCenter PMO event contained an invalid problem number.	Examine the PMO output event mapping and check which field contains the problem number.

Table 9. Error messages (continued)

Error	Description	Action
HP OpenView Writer <i>name</i> : Error in the construction of the Open action SQL.	An error occurred when constructing the open action SQL or the expansion definitions.	Check the SQL syntax and expansion definitions in the open action SQL.
HP OpenView Writer <i>name</i> : Failed to create Event for journal update.	An error occurred in the creation of the ServiceCenter event, due to insufficient resources.	Release memory by stopping unnecessary processes. Increase the machine's available swap space or add more physical memory.
HP OpenView Writer <i>name</i> : Failed to send journal update event to SC.	The gateway sent the journal update event to ServiceCenter.	Examine the ServiceCenter writer module log file for further details on this error.
HP OpenView Writer <i>name</i> : Failed to find the COUNTER_PART attribute for the writer. This is necessary due to bidirectional nature.	The COUNTER_PART attribute is a mandatory attribute that was not specified in the writer definition.	Add the correct attribute to the writer definition and restart the gateway.
HP OpenView Writer <i>name</i> : Is not a name for an ObjectServer reader.	The name given in the FEEDBACK_SERVER attribute was not the name of an object sever reader.	Add the correct attribute to the writer definition and restart the gateway.
HP OpenView Writer <i>name</i> : Reader name was not found for counter part.	The specified counter part reader was not found in the gateway configuration.	Add the correct reader name in the configuration file and restart the gateway.
HP OpenView Writer <i>name</i> : Failed to read the conversions table.	The writer failed to read the conversion table.	Check that the conversion files exists. Check that the file has the correct field definitions.
HP OpenView Writer <i>name</i> : Failed to block on data feed from SC Reader Module.	The gateway to block the data from ServiceCenter.	Examine the ServiceCenter reader module log file for further details about the error.
HP OpenView Writer <i>name</i> : Failed to read the system sequence file. Assuming 0000 start.	The system sequence number could not be read or accessed.	Check existence of the file and that it has read permissions in \$OMNIHOME/var/.
HP OpenView Writer <i>name</i> :: Invalid system sequence number found in the sequence file <i>seq num</i> . Assuming 0 start.	Corrupted sequence file.	Delete the sequence file and restart the gateway.
HP OpenView Writer <i>name</i> : SC Event received with error in the <i>evusrseq</i> field.	A ServiceCenter event was received that contained an invalid value for the <i>evusrseq</i> field.	Check the ServiceCenter format control settings for the generation of output events.

Table 9. Error messages (continued)

Error	Description	Action
HP OpenView Writer name: Failed to find serial <i>serial num</i> in cache for return PMO event.	Received and returned a PMO event sent by the gateway for an opened problem that was not in the cache and could not be obtained again.	No action required.
HP OpenView Writer name: Problem number already determined for serial <i>serial num</i> .	A second return PMO event was received for the same serial number. The problem ticket has already been received.	Configure ServiceCenter to only send one PMO return.
HP OpenView Writer name: Failed to find PM <i>number</i> in cache for return PMU event.	A PMU event was received for a problem that no longer is in the cache and could not be obtained again.	The alert may no longer exist in the ObjectServer. Check the ObjectServer.
HP OpenView Writer name: Update Feedback Failed.	An error occurred while attempting to feed back update details to the ObjectServer.	Examine error messages previous to this message to determine problem cause.
HP OpenView Writer name: Failed to find PM <i>number</i> in cache for return PMC event.	A PMC event was received for a problem that is no longer in the cache and could not be obtained again.	The alert may no longer exist in the ObjectServer. Check the ObjectServer.
HP OpenView Writer name: Close Feedback Failed.	An error occurred while attempting to feed back close details to the ObjectServer.	Examine error messages prior to this message to determine the problem cause.
HP OpenView Writer name: Failed to read SC event from SC Reader Module.	The gateway failed to read a ServiceCenter event from the reader module	Examine the ServiceCenter reader module log file for further details on the error
HP OpenView Writer name: Received event of type event type which was unexpected.	An unexpected event was read by the ServiceCenter writer module.	The writer module should only read PMO, PMU, and PMC event types. The event returned is ignored.
HP OpenView Writer name: Failed to save current eventout system sequence number.	Could not write the current event out sequence number.	Check the write permissions on the sequence file in \$OMNIHOME/var/.
HP OpenView Writer name: Failed to shutdown Reader/Writer Modules.	The gateway failed to shut down the ServiceCenter reader/writer modules.	Examine the ServiceCenter reader/writer module log files for further details on this error.
HP OpenView Writer name: Failed to disconnect feedback connection.	This is a warning message.	No action is required.

Table 9. Error messages (continued)

Error	Description	Action
HP OpenView Writer <i>name</i> : Failed to delete problem ticket from cache for serial <i>serial num</i> .	This is a warning message.	No action is required.

Appendix A. Notices and Trademarks

This appendix contains the following sections:

- Notices
- Trademarks

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, AIX, Tivoli, zSeries, and Netcool® are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.



Part Number:

SC14-7634-03



(1P) P/N: