IBM<sup>®</sup> Tivoli<sup>®</sup> Netcool/OMNIbus Gateway for TSRM 5.0

*Reference Guide April 12, 2018* 



#### Note

Before using this information and the product it supports, read the information in <u>Appendix A</u>, "Notices and Trademarks," on page 47.

#### **Edition notice**

This edition (SC27-2703-10) applies to version 5.0 of IBM Tivoli Netcool/OMNIbus Gateway for TSRM and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC27-2703-09.

#### <sup>©</sup> Copyright International Business Machines Corporation 2009, 2018.

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	viii
Chapter 1. Gateway for TSRM	1
Summary	1
Overview	2
Bidirectional data flow	
XML template files	
Installing the gateway	5
Installing the gateway	5
Installing TAL automations	8
Configuring the gateway server	8
Setting environment variables	8
Configuring the gateway	9
Properties file	
Map definition file	25
Table replication definition file	
Advanced properties	
Configuring SSL connections	
Configuring ICD	
Configuring ICD to receive journal entries from Netcool/OMNIbus events	
Retrieving additional fields from ICD	34
Configuring the ICD event summary field	
Configuring ICD for improved incident reporting performance	
Fine tuning TSRM gateway event processing	
Configuring Netcool/OMNIbus	
Configuring log ticketing functions in Netcool/OMNIbus	
Configuring the Netcool/OMNIbus Event List to use the launch-in-context tool	
Running the gateway	
FIPS mode and encryption	
Testing Netcool/OMNIbus communication with ICD	41
Error messages	
GatewayWatch messages	44
Known issues	
Appendix A. Notices and Trademarks	47
Notices	
Trademarks	

# About this guide

Г

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 TSRM 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/SSSHTQ/omnibus/common/kc\_welcome-444.html? lang=en

Table 1. Document modification history		
Document version	Publication date	Comments
SC27-2703-00	June 30, 2009	First IBM publication.
SC27-2703-01	June 4, 2010	Summary table updated.
		Support for TSRM 7.2 added.
		Updated installation information added.
		Error handling information added to <u>"Advanced properties" on</u> page 29.
		Descriptions for the following properties added to <u>"Properties and command line options" on page 10</u> :
		• Gate.TAL.StateChange FieldName
		• Gate.TSRM.Optional UpdateURLList
		• Gate.TSRM.MapObject StructureSeparator
		• Gate.TSRM. CloseOnDeleteList
		• Gate.TSRM. NonApplication ErrorTitles
		• Gate.TSRM.ErrorCode FieldName
		• Gate.TSRM.Timeout
		Note added to the <b>Gate.TAL.NBScriptFileName</b> property.
		Details about how to add sub-object structures to the incidents created in ICD; see <u>"Adding sub-object structures" on page 27</u> .

© Copyright IBM Corp. 2009, 2018

Table 1. Document modification history (continued)			
Document version	Publication date	Comments	
SC27-2703-02	February 25, 2011	Information about using this guide added in <u>"Conventions used in</u> this guide" on page viii.	
		Summary information updated in <u>"Summary" on page 1</u> .	
		New installation information added in <u>"Installing the gateway" on</u> page 5.	
		Gateway configuration information updated in <u>"Configuring the</u> properties file" on page 9.	
		Added descriptions for the following properties in <u>"Properties and</u> command line options" on page 10:	
		• Gate.TSRM.Attribute TokenSeparator	
		• Gate.TSRM.Optional QueryURLList	
		• Gate.TSRM.Socket Timeout	
		Removed references to the Gate.TAL.StateChange FieldName property.	
		Added new Step 10 to configuration procedure in <u>"Configuring</u> ICD to receive journal entries from Netcool/OMNIbus events" on page 33.	
		ICD event summary configuration information added in "Configuring the ICD event summary field" on page 36.	
		Instructions for running the gateway updated in <u>"Running the</u> gateway" on page 41.	
		Error messages updated in <u>"Error messages" on page 42</u> .	
SC27-2703-03	August 26, 2011	Information about configuring SSL connections added in "Configuring SSL connections" on page 33.	
SC27-2703-04	March 2, 2012	Information about configuring SSL connections updated in "Configuring SSL connections" on page 33.	

Table 1. Document modification history (continued)			
Document version	Publication date	Comments	
SC27-2703-05	September 30, 2012	Descriptions for the following properties added to <u>"Properties</u> and command line options" on page 10:	
		• Gate.TSRM.DateTimeFormat	
		• Gate.TSRM.FilterExclusionList	
		• Gate.TSRM.FilterRepeatedValues	
		• Gate.TSRM.PollTimeIntervalWindow	
		<u>"Polling ICD for unread tickets" on page 31</u> updated with filtering repeated values information.	
		"Removing reliance on the ICD host clock" on page 32 added.	
		Information about configuring SSL connections updated in "Configuring SSL connections" on page 33.	
		<u>"Configuring ICD for improved incident reporting performance"</u> on page 37 added.	
		"Fine tuning TSRM gateway event processing" on page 38 added.	
		"Configuring the Netcool/OMNIbus Event List to use the launch- in-context tool" on page 39	
SC27-2703-06	November 30 ,	"Summary" on page 1 updated.	
	2012	Support for ICD 7.5 added.	
		"Installing the gateway" on page 5 updated.	
		Description for the following properties in <u>"Configuring the</u> properties file" on page 9 updated:	
		• Gate.TSRM.CloseOnDelete	
		• Gate.TSRM.Version	
		"Table replication definition file" on page 28 updated.	
		<u>"Known issues" on page 45</u> added.	
SC27-2703-07	November 8.	"Summary" on page 1 updated.	
	2013	"FIPS mode and encryption" on page 41 updated.	
		Startup command file added to the guide	
		"Table replication definition file" on page 28 undated	
SC27-2703-08	March 12, 2015	<u>"Summary" on page 1</u> updated.	
		Descriptions for the following properties added to <u>"Properties</u> and command line options" on page 10:	
		• Gate TAL TTJournal ServerNameFieldName	
		• Gate.TAL.TTJournalServerSerialFieldName	
		64 bit support added.	
		1	

Table 1. Document modification history (continued)			
Document version	Publication date	Comments	
SC27-2703-09	July 20, 2017	Descriptions for the following properties added to <u>"Properties</u> and command line options" on page 10: • Gate.RdrWtr.IgnoreStatusFilter	
SC27-2703-10	April 12, 2018	Gateway certified against IBM Control Desk V7.5 and V7.6.	

# **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.

Table 2. Directory names for the arch variable		
Operating system	Directory name represented by <i>arch</i>	
AIX <sup>®</sup> systems	aix5	
Red Hat Linux <sup>®</sup> and SUSE systems	linux2x86	

Table 2. Directory names for the arch variable (continued)		
Operating system Directory name represented by <i>arch</i>		
Linux for System z <sup>®</sup>	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.

 ${\bf x}\,$  IBM Tivoli Netcool/OMNIbus Gateway for TSRM: Reference Guide

# **Chapter 1. Gateway for TSRM**

The IBM Tivoli Netcool/OMNIbus Gateway for TSRM provides bidirectional communication between Netcool/OMNIbus and IBM Control Desk.

**Note :** IBM Control Desk formally known as Tivoli Service Request Manager (TSRM), IBM Maximo and IBM SmartCloud Control Desk.

This guide contains the following sections:

- "Summary" on page 1
- "Overview" on page 2
- "Installing the gateway" on page 5
- "Configuring the gateway" on page 9
- "Configuring SSL connections" on page 33
- "Configuring ICD" on page 33
- "Configuring Netcool/OMNIbus" on page 39
- "Running the gateway" on page 41
- "Testing Netcool/OMNIbus communication with ICD" on page 41
- "Error messages" on page 42
- "GatewayWatch messages" on page 44

## Summary

Each gateway works in a different way to provide an interface with the ObjectServer. Use this summary information to learn about the Gateway for TSRM.

The following table provides a summary of the gateway:

Table 3. Summary		
Gateway target	IBM Control Desk V7.5 and V7.6	
Gateway executable file name	nco_g_tsrm	
Gateway installation package	omnibus- <i>arch</i> -gateway-nco_g_tsrm- <i>version</i>	
Package version	5.0	
Gateway supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: <u>https://www-304.ibm.com/support/docview.wss?</u> uid=swg21610165	

Table 3. Summary (continued)		
Configuration files	<pre>\$OMNIHOME/etc/NCO_GATE.props</pre>	
	<pre>\$OMNIHOME/gates/tsrm/tsrm.map</pre>	
	<pre>\$OMNIHOME/gates/tsrm/tsrm.script</pre>	
	<pre>\$OMNIHOME/gates/tsrm/tools.sql</pre>	
	<pre>\$OMNIHOME/gates/tsrm/tsrm.rdrwtr.tblrep.def</pre>	
	<pre>\$OMNIHOME/gates/tsrm/create.xml.template</pre>	
	<pre>\$OMNIHOME/gates/tsrm/journal.xml.template</pre>	
	<pre>\$OMNIHOME/gates/tsrm/query.xml.template</pre>	
	<pre>\$OMNIHOME/gates/tsrm/update.xml.template</pre>	
Additional binaries	nco_g_tsrm.jar	
Requirements	IBM® Tivoli® Netcool/OMNIbus V7.4.0. (and above)	
	Java <sup>™</sup> Gateway Infrastructure library: libngjava-3_0	
	Ticketing Abstraction Layer (TAL) library: libtal-5_0	
	<b>Note :</b> libngjava and libtal require one of the following versions of Netcool/OMNIbus:	
	• V7.4.0	
	<b>Note :</b> On the Windows operating system a minimum of Java 1.5 is required.	
Failover/failback functionality	Available	
Multicultural support	Available	
	For information about configuring multicultural support, including language options, see the <i>IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide</i> .	
	<b>Note :</b> Installations of the gateway on Windows operating systems do not support UTF-8 character encoding. For some languages, such as Traditional Chinese, the gateway is therefore unable to correctly pass events from the ObjectServer to ICD. UTF-8 character encoding is available for installations of the gateway on UNIX and Linux operating systems.	
IP environment	IPv4 and IPv6	
Federal Information Processing Standards (FIPS)	IBM Tivoli Netcool/OMNIbus uses the FIPS 140-2 approved cryptographic provider: IBM Crypto for C (ICC) certificate 384 for cryptography. This certificate is listed on the NIST web site at <u>http:// csrc.nist.gov/cryptval/140-1/1401val2004.htm</u> . For details about configuring Netcool/OMNIbus for FIPS 140-2 mode, see the <i>IBM</i> <i>Tivoli Netcool/OMNIbus Installation and Deployment Guide.</i>	

## **Overview**

IBM Control Desk (ICD) is an integrated service management solution. ICD can function as a service desk for both internal IT assets and internal non-IT assets (such as facilities or fleet). ICD helps to improve IT

performance by providing automation of processes and better visibility of service support functions, commitments, and measurements.

The Gateway for TSRM is a bidirectional gateway that allows you to create, update, and close tickets (either incidents or service requests) in ICD, using XML over HTTP. The gateway allows you to create and maintain ICD tickets from Netcool/OMNIbus events, either manually (using an ICD user interface within Netcool/OMNIbus) or automatically (using scripts). There is a one-to-one mapping between a single Netcool/OMNIbus event and a single ICD ticket.

The gateway supports the following functions:

- Flexible filtering that allows you to specify which event details are sent from Netcool/OMNIbus to ICD.
- Mapping of Netcool/OMNIbus alerts.status fields to any fields in the ICD schema.
- Assignment of literal values (for example, varchar, int, date) to specified fields in the ICD schema.
- Conversion of data types.
- Failover between Netcool/OMNIbus and ICD.
- Failback between Netcool/OMNIbus and ICD can be configured using the **Gate.RdrWtr.FailbackEnabled** property.
- Sub-object structures within ICD.
- Reporting within Netcool/OMNIbus of ICD operation failures.
- Separate URLs for creating new incidents and updating existing incidents.
- Mapping of Netcool/OMNIbus severities to any ICD priorities, for more information see <u>"Mapping</u> attributes" on page 26
- Closing tickets in ICD when the associated event in Netcool/OMNIbus is deleted can be configured by specifying the **Gate.TSRM.CloseOnDeleteList** property.
- Creation of journal entries, for more information see <u>"Configuring ICD to receive journal entries from</u> Netcool/OMNIbus events" on page 33.

To facilitate bidirectional operation, the gateway uses ICD's built-in integration framework and the Ticket Abstraction Layer (TAL) library. In the context of this integration, the gateway is regarded as an external application. The TAL library is supplied as a separate patch. It provides the infrastructure required to generate ICD tickets from Netcool/OMNIbus events and to exchange ticket data with ICD.

**Note :** Not all fields have bidirectional data flow enabled by default. You can add bidirectional support to additional fields, for more information see <u>"Mapping attributes" on page 26</u> and <u>"Sending data from ICD</u> to the ObjectServer" on page 28.

**Note :** The gateway can generate either incidents or service requests, but not both. If you want to create both incidents and service requests from Netcool/OMINbus events, you must run two instances of the gateway. The bidirectional transfer of journals is not currently supported. Journal entries within the ObjectServer are created in ICD as work log entries, but work log entries are not forwarded from ICD to the ObjectServer as journals.

#### **Bidirectional data flow**

The Gateway for TSRM enables bidirectional flow of data between the ObjectServer and ICD.

The following diagram shows the flow of data between the ObjectServer and ICD.



#### Figure 1. Data flow between the ObjectServer and ICD

The gateway can create an ICD ticket from an event in the Netcool/OMNIbus Event List by mapping fields in the ObjectServer to fields in ICD.

Once a ticket is created, you can add a journal to the original event in the ObjectServer. The gateway passes the data to the ICD where it is converted into a work log.

If the status of the ticket within ICD changes, the gateway passes the changed status back to the ObjectServer.

#### XML template files

The gateway is supplied with a set of XML template files that determine how the gateway exchanges data between Netcool/OMNIbus and ICD.

The XML template files define how the gateway performs the following functions:

- Creating ICD tickets from Netcool/OMNIbus events.
- Querying ICD for ticket details.
- Updating existing ICD tickets.
- Creating ICD work logs from Netcool/OMNIbus journal entries.

Before you can use the XML template files, you must add the structure for incident and work log objects to the ICD object structure using IBM Control Desk. For details of how to do this, see <u>"Configuring ICD to</u> receive journal entries from Netcool/OMNIbus events" on page 33.

If your implementation of ICD uses sub-object structures (for example, to associate one or more assets with a ticket), you can define the sub-object structure that the gateway must add to incidents created within Netcool/OMNIbus by customizing the map definition file. This adds the appropriate code to the XML template files. For details, see "Adding sub-object structures" on page 27.

# Installing the gateway

The Gateway for TSRM and its related patches are installed as described below.

To install the Gateway for TSRM, use the following steps:

- 1. Install the gateway.
- 2. Configure the gateway server.
- 3. Set environment variables.

### 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**.
- 6 IBM Tivoli Netcool/OMNIbus Gateway for TSRM: Reference Guide

## 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:

- <u>"The installation wizard" on page 7</u>
- 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 reponse.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.

## **Installing TAL automations**

The Ticket Abstraction Layer (TAL) library provides the infrastructure required to generate ICD tickets from Tivoli Netcool/OMNIbus events and to exchange ticket data with ICD. After the TAL library is installed by default with Tivoli Netcool/OMNIbus, you must install the TAL automations. The installation steps are the same regardless of which version of Tivoli Netcool/OMNIbus you are using.

To install the TAL automations on UNIX and Linux operating systems, run the following commands using the SQL interactive interface utility (nco\_sql):

```
nco_sql -U username -P password -S obj_serv < $OMNIHOME/gates/tal/
tal_automations.sql
```

nco\_sql -U username -P password -S obj\_serv < \$OMNIHOME/gates/tsrm/tsrm.sql</pre>

where:

- username is the name of the user authorized to run Netcool/OMNIbus.
- *password* is the password associated with this user.
- *obj\_serv* is the name of ObjectServer instance to which the gateway will connect.

To install the TAL automations on Windows operating systems, copy the contents of the following files into the Tivoli Netcool/OMNIbus Administrator SQL interactive interface window and click the **Execute SQL** button:

```
%OMNIHOME%\gates\tal\tal_automations.sql
```

%OMNIHOME%\gates\tsrm\tsrm.sql

**Note :** For information about using the Tivoli Netcool/OMNIbus Administrator interface and the SQL interactive interface utilities, see the *IBM Tivoli Netcool/OMNIbus Administration Guide*.

## Configuring the gateway server

You must create the gateway server in the Netcool/OMNIbus interfaces file.

Use the Server Editor (nco\_xigen) to specify the hostname and port number of the gateway host in the Netcool/OMNIbus interfaces file. If you want to run multiple gateways, you must use a different name for each instance.

Use the following hostname for the primary gateway server:

GAT\_TSRM

**Note :** For information about using the Server Editor, see the *IBM Tivoli Netcool/OMNIbus Administration Guide*.

## **Setting environment variables**

Environment variables are specific preset values that establish the working environment of the gateway. From the environment variable specified, the gateway receives path information for the directories in which library files are present.

**Note :** This information only applies to gateway installations on Windows operating systems. If you are installing the gateway on a UNIX or Linux operating system, you may skip this procedure.

On Windows 2008 add the following directories to the %LIBPATH% environment variable:

- %NCHOME%\platform\win32\jre\_1.5.6\jre\bin
- %NCHOME%\platform\win32\jre\_1.5.6\jre\bin\classic

# **Configuring the gateway**

The gateway uses centralized property management, which separates properties from data processing configuration.

Use the properties file to configure the gateway properties. Use the map definition and table replication definition files to configure data processing.

**Note :** You must have write access permissions to the directory where you want to run the gateway or you will get an error message referring to derby.log or 'FileNotFoundException'. For more information see "Known issues" on page 45.

The following topics describe the configuration files and how to use them:

- "Properties file" on page 9
- "Map definition file" on page 25
- "Advanced properties" on page 29

#### **Properties file**

The properties file is a text file that contains a set of properties and their corresponding values. These properties define the operational environment of the gateway, such as connection details and the location of the other configuration files.

### Configuring the properties file

Before running the gateway for the first time, you must rename the installed properties file and specify some basic properties.

To configure the properties file, use the following steps:

- 1. Rename the properties file that came with the installation package:
  - a. Copy the NCO\_GATE.props file to the \$OMNIHOME/etc directory.
  - b. Change the name of the copied file from NCO\_GATE.props to GAT\_TSRM.props.
- 2. After renaming the properties file, specify the following property values in the new GAT\_TSRM.props file:

Property	Value
MessageLog	<pre>\$OMNIHOME/log/GAT_TSRM.log</pre>
Name	GAT_TSRM
PropsFile	<pre>\$0MNIHOME/etc/GAT_TSRM.props</pre>

3. If you have Java security enabled, use the **Gate.TSRM.UserName** and **Gate.TSRM.Password** properties to specify the user name and password required to access ICD.

Note : If you do not have Java security enabled, you do not need to specify values for these properties.

4. Edit the remaining values in the properties file to suit your operating environment.

## **Properties and command line options**

You use properties to define the operational environment of the gateway. You can override the default values by editing the properties file or by using the command line options.

The following tables describe the properties available with the Gateway for TSRM. For more information about generic and Inter-Process Communication (IPC) properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

The following sections describe the properties used to configure the gateway:

- "Common Tivoli Netcool/OMNIbus properties" on page 10
- "Generic gateway properties" on page 12
- <u>"Java properties" on page 13</u>
- <u>"Mapping properties</u>" on page 13
- "Gateway Reader-Writer properties" on page 14
- <u>"Ticket Abstraction Layer (TAL) properties" on page 16</u>
- "Gateway-specific properties" on page 18

#### **Common Tivoli Netcool/OMNIbus properties**

Table 4 on page 10 lists the available common properties.

Table 4. Common Tivoli Netcool/OMNIbus properties		
Property name	Command line option	Description
ConfigCryptoAlg string	-configcryptoalg <i>string</i>	Use this property to specify the encryption algorithm that the gateway uses. The default is AES.
ConfigKeyFile string	-configkeyfilestring	Use this property to specify the encryption key used with the encrypted data. The default is "".
<b>Connections</b> integer	-connections integer	Use this property to specify the maximum number of client connections that can be made to the gateway server. The default is 30.
<b>Help</b> boolean	-help <i>boolean</i>	Use this property to instruct the gateway to display application help information on startup and exit. The default is FALSE.

Table 4. Common Tivoli Netcool/OMNIbus properties (continued)			
Property name	Command line option	Description	
MaxLogFileSize integer	-maxlogfilesize <i>integer</i>	Use this property to specify the size (in bytes) that the gateway allocates for the log file. When the log file reaches this size, the gateway renames the log file by appending the suffix .old and creates a new log file. The default is 1024.	
MessageLevel string	-messagelevel string	Use this property to specify the reporting level of the log file messages. The default is warn.	
MessageLog string	-messagelog <i>string</i>	Use this property to specify the location of the message log file. The default is \$OMNIHOME/log/ NCO_GATE.log.	
Name string	-name string	Use this property to specify the name of the current gateway instance. If you want to run multiple gateways on one machine, you must use a different name for each instance. The default is GAT_TSRM.	
Props.CheckNames boolean	No command line equivalent.	Use this property to instruct the gateway to shut down if any property in the properties file is set to an invalid value. The default is TRUE.	
<b>PropsFile</b> string	-propsfile string	Use this property to specify the location of the gateway properties file. The default is \$OMNIHOME/etc/ NCO_GATE.props.	
UniqueLog boolean	-uniquelog <i>boolean</i>	Use this property to specify that log file names are made unique by adding the Process ID (PID) of the gateway to the file name. The default is FALSE.	

Table 4. Common Tivoli Netcool/OMNIbus properties (continued)		
Property name	Command line option	Description
<b>Version</b> boolean	-version boolean	Use this property to instruct the gateway to display information about the application version on startup and shutdown. The default is FALSE.

## **Generic gateway properties**

Table 5 on page 12 lists the available generic gateway properties.

Table 5. Generic gateway properties		
Property name	Command line option	Description
Gate.CacheHashTblSize integer	-cachehtblsize <i>integer</i>	Use this property to specify the number of elements the gateway allocates for the hash table cache.
Gate.MapFile string	-mapfile string	Use this property to specify the mapping file for the gateway to use.
		The default is \$OMNIHOME/ gates/tsrm/tsrm.map.
Gate.NGtkDebug boolean	-ngtkdebug <i>boolean</i>	Use this property to enable the logging of NGTK library debug messages.
		The default is TRUE.
Gate.PAAware integer	-paaware integer	This property indicates whether the gateway is PA aware.
		The default is 0 (not PA aware).
		<b>Note :</b> This property is maintained by the PA server and is included in the properties file for information only.
Gate.PAAwareName string	-paname string	This property indicates the name of the PA controlling the gateway.
		The default is "".
		<b>Note :</b> This property is maintained by the PA server and is included in the properties file for information only.

Table 5. Generic gateway properties (continued)		
Property name	Command line option	Description
Gate.UnixAdminGrp string	-unixadmingrp <i>string</i>	Use this property to specify the administration group to which the gateway must belong if standard UNIX authentication is used. The default is ncoadmin.
Gate.UsePamAuth boolean	-usepamauth <i>boolean</i>	Use this property to specify whether PAM authentication is used. The default is FALSE. <b>Note :</b> To run the gateway in FIPS 140-2 mode, you must set this property to TRUE.

### **Java properties**

Table 6 on page 13 lists the available Java properties.

Table 6. Java properties		
Property name	Command line option	Description
Gate.Java.Arguments string	-javaarguments <i>string</i>	Use this property to specify the arguments to use when starting Java. The default is "".
Gate.Java.ClassPath string	-javaclasspath <i>string</i>	Use this property to specify the environment variable used to store the location of the Java libraries. The default is \$CLASSPATH.
Gate.Java.Debug boolean	-javadebug <i>boolean</i>	Use this property to enable the logging of Java debug messages. The default is TRUE.
<b>Gate.Java.LibraryPath</b> string	-javalibrarypath <i>string</i>	Use this property to specify the location of the Java libraries that will be set in the environment variable specified by the <b>Gate.Java.ClassPath</b> property. The default is "".

# **Mapping properties**

Table 7 on page 14 lists the available mapping properties.

Table 7. Mapping properties			
Property name	Command line option	Description	
Gate.Mapper.Debug boolean	-mapperdebug <i>boolean</i>	Use this property to enable the logging of mapper debug messages. The default is TRUE.	
Gate.Mapper. ForwardHistoricJournals boolean	-mapperforhistjrnl <i>boolean</i>	Use this property to specify whether the gateway forwards all historic journals on converted update. The default is FALSE.	

## **Gateway Reader-Writer properties**

Table 8 on page 14 lists the available gateway reader-writer properties.

Table 8. Gateway Reader-Writer properties			
Property name	Command line option	Description	
Gate.RdrWtr.Debug boolean	-debug <i>boolean</i>	Use this property to enable the logging of gateway reader debug messages. The default is TRUE.	
Gate.RdrWtr.Description string	-description <i>string</i>	Use this property to specify the application description for the reader connection. This description is used in triggers and allows you to determine which component of the gateway attempted to perform an action. The default is ticket_gate.	
Gate.RdrWtr. FailbackEnabled boolean	-readerfailbackenabled boolean	Use this property to specify whether the gateway attempts to fail back to the primary ObjectServer following a ObjectServer failover. The default is FALSE. <b>Note :</b> The gateway attempts to fail back with the frequency specified by the <b>Gate.RdrWtr.</b> <b>FailbackTimeout</b> property.	

Table 8. Gateway Reader-Writer properties (continued)			
Property name	Command line option	Description	
Gate.RdrWtr. FailbackTimeout integer	-readerfailbacktimeout integer	Use this property to specify the frequency (in seconds) with which the gateway attempts to fail back to the primary system following a system failover.	
		The default is 30 seconds.	
		Note : The gateway attempts to fail back to the primary ObjectServer only if the Gate.RdrWtr. FailbackEnabled property is set to TRUE.	
Gate.RdrWtr.IducFlushRate integer	-iducflushrate integer	Use this property to specify the rate (in seconds) of the granularity of the reader.	
		If you set this property to 0, the reader gets its updates at the same granular rate as that of the ObjectServer to which it is connected.	
		The default is 0.	
		<b>Note :</b> If you set this property to a value greater than 0, the reader issues automatic IDUC flush requests to the ObjectServer with this frequency. This enables the reader to run at a faster granularity than that of the ObjectServer, thus enabling the gateway to capture more detailed event changes in systems where the ObjectServer itself has high granularity settings.	
Gate.RdrWtr. IgnoreStatusFilter	-ignorestatusfilter boolean	Use this property to enable or disable the passing through of journal entries from the ObjectServer to IBM Control Desk. This property takes the following values: FALSE: The gateway passes through journal entries	
		TRUE: The gateway ignores the status filter, and so does not pass through journal entries.	
		The default is FALSE.	

Table 8. Gateway Reader-Writer properties (continued)		
Property name	Command line option	Description
<b>Gate.RdrWtr.LogOSSql</b> boolean	-logossql boolean	Use this property to specify whether the gateway logs all SQL commands sent to the ObjectServer in debug mode. The default is FALSE.
Gate.RdrWtr.Password string	-password string	Use this property to specify the password associated with the user specified by the <b>Gate.RdrWtr.Username</b> property. The default is " ". <b>Note :</b> You must encrypt this password using the nco_g_crypt utility. For more information, see the <i>IBM Tivoli</i> <i>Netcool/OMNIbus Administration</i> <i>Guide</i> .
Gate.RdrWtr. ReconnectTimeout integer	-reconntimeout <i>integer</i>	Use this property to specify the time (in seconds) between each reconnection poll attempt that the gateway makes if the connection to the ObjectServer is lost. The default is 30 seconds.
Gate.RdrWtr.Server string	-server string	Use this property to specify the name of the ObjectServer from which the gateway reads alerts The default is NCOMS.
Gate.RdrWtr. TblReplicateDefFile string	-tblrepdeffile <i>string</i>	Use this property to specify the path to the table replication definition file. The default is \$OMNIHOME/ gates/tsrm/ tsrm.rdrwtr.tblrep.def.
Gate.RdrWtr.Username string	-username string	Use this property to specify the user name used to authenticate the ObjectServer connection. The default is root.

## Ticket Abstraction Layer (TAL) properties

Table 9 on page 17 lists the available TAL properties.

Table 9. TAL properties		
Property name	Command line option	Description
Gate.TAL.ForwardClear UpdatesOnly boolean	-forwardclearupdates boolean	Use this property to specify whether the gateway only sends update events to ICD if they are clear events. The default is FALSE.
<b>Gate.TAL.NBScriptFileName</b> string	-nbscriptfilename string	Use this property to specify the name of the scripting file used for sending notification data to the ObjectServer. The default is \$OMNIHOME/ gates/tsrm/tsrm.script. <b>Note :</b> If you want to stop the gateway polling requests, you must leave this property blank.
<b>Gate.TAL.TTFilterClause</b> string	-ttfilterclause string	Use this property to specify the filter clause used in the table replication definition file for the status table. The default is LogTicket=1. <b>Note :</b> You must specify this value in the form of the SQL condition of the WITH FILTER clause. For example, if the definition file uses the filter WITH FILTER LogTicket=1, set this property to LogTicket =1.
Gate.TAL.TTJournal ChronoFieldName string	-ttjournalchronofieldname string	Use this property to specify the target field name to use in the map for the journal Chrono field. The default is Chrono.
Gate.TAL.TTJournal ServerNameFieldName string	-ttjournalservername fieldname <i>string</i>	Use this property to specify the target field name to use in the map for the journal ServerName field. The default is ServerName.
Gate.TAL.TTJournal ServerSerialFieldName string	-ttjournalserverserial fieldname <i>string</i>	Use this property to specify the target field name to use in the map for the journal ServerSerial field. The default is ServerSerial.

Table 9. TAL properties (continued)		
Property name	Command line option	Description
Gate.TAL.TTNumber FieldNamestring	-ttnumberfieldname string	Use this property to specify the name of the field in the ObjectServer alerts.status table that stores the trouble ticket number.
		The default is TTNumber.
		<b>Note :</b> You will need to create the field that you specify with this property within Tivoli Netcool/ OMNIbus. For details about how to do this, see <u>"Adding a field to the ObjectServer" on page 35</u> .
<b>Gate.TAL.TTStateFieldName</b> string	-ttstatefieldname <i>string</i>	Use this property to specify the name of the integer field in the ObjectServer alerts.status table that stores the trouble ticket state. The default is TTState.

### **Gateway-specific properties**

Table 10 on page 18 lists the available gateway-specific properties.

Table 10. Gateway-specific properties		
Property name	Command line option	Description
Gate.TSRM.Attribute TokenSeparator string	-attributetokenseparator string	Use this property to specify a default separator for naming XML attribute-based tokens when processing the northbound ICD response. The default is _ATTR

Table 10. Gateway-specific properties (continued)		
Property name	Command line option	Description
Gate.TSRM. CloseOnDeleteList string	-closeondeletelist string	Use this property to specify whether a ticket in ICD is closed if the Tivoli Netcool/OMNIbus event for which it was originally created is deleted.
		Specify the values for this property as comma-delimited name-value pairs in the following format: "Name1='Value1',, Nam eN='ValueN'". For example:
		"CLASS='INCIDENT',
		STATUS='RESOLVED'"
		The default is "". The function is not enabled by default.
Gate.TSRM. CreateTicketTemplate string	- tsrmcreatetickettemplate <i>string</i>	Use this property to specify the name of the XML template used to create the ticket within ICD. The default is \$OMNIHOME/ gates/tsrm/ create.xml.template.
<b>Gate.TSRM.DateTimeFormat</b> string	-tsrmdatetimeformat <i>string</i>	Use this property to specify the date time format the gateway uses to process ticket updates sent by ICD to the gateway. The default is yyyy-MM-dd\'T \'HH:mm:ssZ.

Table 10. Gateway-specific properties (continued)		
Property name	Command line option	Description
Gate.TSRM.ErrorCode FieldName string	-errorcodefieldname string	Use this property to specify the ObjectServer field to which the gateway sends the error code that identifies the nature of the ICD error. The <b>Gate.TSRM.ErrorCode</b> <b>FieldName</b> property is used in conjunction with the <b>Gate.TSRM.NonApplication</b> <b>ErrorTitles</b> property to work out what category the errors belong to. Any ICD error responses that match the regular expression in the <b>Gate.TSRM.NonApplication</b> <b>ErrorTitles</b> property will map to error code 1. This property takes the following values: 0: the operation is successful. 1: the error is due to an intermittent issue, such as the server being down, so the operation can be retried. 2: the error is such that the operation cannot be retried, such as an invalid field value. The default is TSRMErrorCode.
Gate.TSRM. FilterRepeatedValues boolean	- tsrmfilterrepeatedvalues <i>boolean</i>	Use this property to specify whether the gateway sends only event data that has changed since the last polling. The filtering will be based on the mapped values in the tsrm.map file and will omit the mapped values specified by the <b>Gate.TSRM.</b> <b>FilterExclusionList</b> property. The default is TRUE.
Gate.TSRM. FilterExclusionList string	-tsrmfilterexclusionlist string	Use this property to specify a list of properties that are not filtered by the gateway if the <b>Gate.TSRM.</b> <b>FilterRepeatedValues</b> is set to true. The default is 'TTNumber'.

Table 10. Gateway-specific properties (continued)		
Property name	Command line option	Description
<b>Gate.TSRM.JournalTemplate</b> <i>string</i>	-tsrmjournaltemplate string	Use this property to specify the name of the XML template used to create work logs in ICD.
		The defaultis \$OMNIHOME/ gates/tsrm/ journal.xml.template.
Gate.TSRM.Language string	-tsrmlanguage <i>string</i>	Use this property to specify the language code for the language used by ICD to process incoming data. The default is EN.
Gate.TSRM.MapObject StructureSeparator string	-mapobjectstructure separator <i>string</i>	Use this property to specify the separator that the gateway uses between sub-objects when generating the XML code that it sends to ICD to create or update tickets. See <u>"Adding sub-object</u> <u>structures" on page 27</u> for more information. The default is "".
<b>Gate.TSRM.MaxRetries</b> integer	-tsrmmaxretries integer	Use this property to specify the number of times that the gateway retries polling ICD after the occurrence of an error, an HTTP timeout, or a socket-related exception, before shutting down. The default is 5.
Gate.TSRM.NonApplication ErrorTitles string	- nonapplicationerrortitle s <i>string</i>	Use this property to specify a comma-delimited list of application error titles (sent by ICD) that the gateway should interpret as connection errors. The default is .*500 Internal Server Error.*
Gate.TSRM.Optional UpdateURLList string	-optionalupdurllist string	If your ICD implementation requires two URLs (one for creating tickets and one for updating tickets), use this property to specify the URL needed to communicate with ICD for <b>updating</b> tickets. Otherwise, leave this property blank. The default is "".

Table 10. Gateway-specific properties (continued)		
Property name	Command line option	Description
Gate.TSRM.Password string	-tsrmpassword string	Use this property to specify the ICD password associated with the user name specified by the <b>Gate.TSRM.Username</b> property. The default is maxadmin.
Gate.TSRM.PollTime integer	-tsrmpolltime integer	Use this property to specify the frequency (in seconds) with which the gateway polls ICD. The default is 30 seconds.
Gate.TSRM. PollTimeIntervalWindow integer	-tsrmpolltime intervalwindow <i>integer</i>	Use this property to specify a different CHANGEDATE format for the gateway to use to determine new events in ICD. The default is -1 : This function is disabled.
Gate.TSRM. QueryRetrieveList string	-tsrmqueryretrievelist string	Use this property to specify a comma-separated list of ICD ticket attributes that the gateway sends back to the ObjectServer for further processing. The default is STATUS. <b>Note :</b> If you add any ICD fields to this list, you must add the XML structure for that field to that of the ICD incident or service request. For details of how to do this, see <u>"Retrieving additional</u> fields from ICD" on page 34.
Gate.TSRM. QueryTicketTemplate string	-tsrmquerytickettemplate string	Use this property to specify the name of the XML template used to query the ticket within ICD. The default is \$OMNIHOME/ gates/tsrm/ query.xml.template.
Gate.TSRM.RetryInterval integer	-tsrmretryinterval integer	Use this property to specify the interval (in seconds) between successive polls of the ICD. The default is 15 seconds.

Table 10. Gateway-specific properties (continued)		
Property name	Command line option	Description
<b>Gate.TSRM.SocketTimeout</b> <i>integer</i>	-tsrmsockettimeout <i>integer</i>	Use this property to specify the length of time (in seconds) that the gateway waits for a response to a requested operation from ICD. The default is 15 seconds.
Gate.TSRM.SourceName string	-tsrmsourcename string	Use this property to specify the name used to tag incidents and work log objects. This allows you to indicate that the incident was created by the gateway. The default is NETCOOL.
Gate.TSRM.StatusFieldName string	-tsrmstatusfieldname <i>string</i>	Use this property to specify the name of the mapper field used to populate the STATUS workflow attribute in ICD tickets. The default is STATUS. <b>Note :</b> If the value of the mapper field is empty, the alert is not forwarded to the ICD server. This is useful in a scenario where you might want to set the alert to RESOLVED if the severity of the alert is clear (or for any other reason you may not want to forward it). That is why the following lookup definition is included in the tsrm.map file: CREATE LOOKUP StatusTable ( {0, 'RESOLVED'}) DEFAULT= '';
<b>Gate.TSRM.ThreadPoolSz</b> integer	-tsrmthreadpoolsz integer	Use this property to specify the number of concurrent threads that the gateway uses to communicate with ICD. The default is 5. <b>Note :</b> In most cases, you will not need to change this value.

Table 10. Gateway-specific properties (continued)			
Property name	Command line option	Description	
Gate.TSRM. TicketObjectName string	-tsrmticketobjectname string	Use this property to specify the object name for the ICD incident or service request object.	
		The default is MXINCIDENT.	
		<b>Note :</b> If you are using the gateway to generate service requests instead of incidents, set this property to MXSR.	
Gate.TSRM. TicketObjectType string	-tsrmticketobjecttype string	Use this property to specify the object type created by the gateway within ICD.	
		The default is INCIDENT.	
		<b>Note :</b> If you are using the gateway to generate service requests instead of incidents, set this property to SR.	
Gate.TSRM.TimeZone string	-tsrmtzdiff <i>string</i>	Use this property to specify the timezone used by the ICD server relative to GMT. For example, to specify four hours behind GMT, set this property to GMT-4.	
		The default is " ".	
		<b>Note :</b> If the gateway is being run in the same timezone as the ICD server, leave this property blank.	
Gate.TSRM. UpdateTicketTemplate string	- tsrmupdatetickettemplate string	Use this property to specify the name of the XML template used to update the ticket within ICD.	
		The defaultis \$OMNIHOME/ gates/tsrm/ update.xml.template.	
Gate.TSRM.URLList string	-tsrmurllist <i>string</i>	Use this property to specify a comma-delimited string consisting of the URLs required to communicate with ICD.	
		The default is http:// <targethost>/meaweb/os/ MXINCIDENT.</targethost>	
		<b>Note :</b> If you are using the ICD server failover function, you must specify each URL separated by commas.	

Table 10. Gateway-specific properties (continued)		
Property name	Command line option	Description
Gate.TSRM.UserName string	-tsrmusername string	Use this property to specify the name of the user authorized to log on to ICD. The default is maxadmin.
Gate.TSRM.Version string	-tsrmversion <i>string</i>	Use this property to specify the version of ICD with which the gateway can communicate. The default is 7.5.
Gate.TSRM. WorkLogObjectName string	-tsrmworklogobjectname string	Use this property to specify the object name of the ICD work log object name. The default is WORKLOG.

### Map definition file

Mapping defines how the gateway replicates tables by assigning data to appropriate fields in the ObjectServer. The map definition file defines how the gateway maps data received from the ICD to the Status, and Journal tables within the ObjectServer.

To specify the map definition file, use the **Gate.MapFile** property. The default map definition file, tsrm.map, is located in the following directory:

#### \$OMNIHOME/gates/tsrm/

**Note :** If you are upgrading from version 1.1.0 of the gateway to version 1.2.0 (or later), you must populate the CLASS field of incidents with the value INCIDENT by adding the following entries to the CREATE MAPPING StatusMap and CREATE MAPPING JournalMap sections of the map definition file:

```
CREATE MAPPING StatusMap
( 'CLASS' = 'INCIDENT',
...
);
CREATE MAPPING JournalMap
( 'CLASS' = 'INCIDENT',
...
);
```

#### **Mapping syntax**

You can configure the mapping functions of the gateway by using the mapper attributes.

Mappings for use with the ObjectServer writer must use the following syntax:

```
CREATE MAPPING mappingname ( ' dest_fieldname' = ( '@src_fieldname' |
simple_expression | attribute ) [ ON INSERT ONLY ] [ CONVERT TO type ]
[ NOT NULL ('@src_fieldname')] [ , 'dest_fieldname' =
    ( '@src_fieldname' | simple_expression | attribute ) [ ON INSERT ONLY ]
[ CONVERT TO type ] [ NOT NULL ('@src_fieldname')] ] ... );
```

where:

```
• mappingname is the name of the mapping to be created.
```

- dest\_fieldname is the name of the field to be written in ICD.
- *src\_fieldname* is the name of a field in the ObjectServer alerts.status table.
- *simple\_expression* is an integer or a set of integers and operators.
- *attribute* is an attribute name.

The ON INSERT ONLY command is optional and only applies when setting the value of variables. It controls the updating of the field during the life of the alert. When ON INSERT ONLY is included, the field is created once for the alert but is never updated. When ON INSERT ONLY is omitted, the field is updated for any change in the state of the alert.

The CONVERT TO *type* command enables forced conversion for situations where the data type of a source field may not match the data type of a destination field. The data type can be INTEGER, STRING, or DATE.

The NOT NULL command indicates that the mapping is only performed if the source field is not null (it must have a value).

#### **Mapping attributes**

You use attribute names to include additional data in mapping definitions.

Mapping attributes and advanced mapping topics are explained in the following sections:

- "Mapping example" on page 26
- "Modifying the default mapping" on page 27
- "Adding sub-object structures" on page 27
- "Sending data from ICD to the ObjectServer" on page 28

#### **Mapping example**

The following example shows the mappings for the ObjectServer tables into which the gateway writes:

Within this map, @TTNumber and @Chrono are used to provide ticket feedback and to enable failover respectively. They also correspond to, and must be the same as, the values set for the **Gate.TAL.TTNumberFieldName** and **Gate.TAL.TTJournalChronoFieldName** properties. In most

cases you will not need to change these field names but if you do change them you must also change their corresponding values in the properties file.

#### Modifying the default mapping

You can modify the default mapping to send additional fields from Netcool/OMNIbus. The gateway supports both flexible mappings and lookup tables.

For most fields that you want to add to those mapped to ICD, you simply add a new item to the default map definition file.

For example, if you want to populate the REPORTEDEMAIL field of the incident with the ContactName ObjectServer field, add the following line to the StatusMap section of the map definition file:

'REPORTEDEMAIL' = '@ContactName',

The gateway will take the new item and convert it to an XML token that is sent across to the ICD server. These are regarded as flexible mappings.

Certain values (those that use lookup tables) can only be mapped successfully if they are first configured to exist in ICD. This applies to the following fields (among others):

- OWNER
- OWNERGROUP
- CLASSIFICATION ID

For details of the ICD fields that use lookup tables and how to configure them, refer to the *IBM Tivoli SRM Administration Guide*.

#### Adding sub-object structures

If your implementation of ICD uses substructures (for example, to associate one or more assets with a ticket), you can define the sub-object structure that the gateway must add to incidents created within Netcool/OMNIbus by customizing the map definition file.

For example, if your implementation includes a sub-object structure for asset information, and for each asset you can specify its type and asset number, you can add the following map definition to the map definition file:

```
'MULTIASSETLOCCI.RECORDCLASS' = 'INCIDENT' ON INSERT ONLY,
'MULTIASSETLOCCI.ASSETNUM' = '7200' ON INSERT ONLY
```

You must then use the **Gate.TSRM.MapObjectStructureSeparator** property to specify the separator that the gateway inserts between each asset associated with an incident. For example '\\\.'

In the following example, the gateway produces XML for new incidents created from Netcool/OMNMIbus events (the sub-object structure is shown in bold):

```
<CreateMXINCIDENT
   xmlns="http://www.ibm.com/icd"
creationDateTime="2010-03-09T10:14:08+00:00"
   transLanguage="EN"
   messageID="123"
   icdVersion="7.5">
  <MXINCIDENTSet>
    <INCIDENT action="Add"><CLASS><![CDATA[INCIDENT]]&gt;</CLASS>
     <Pre>
     <REPORTEDPRIORITY><![CDATA[4]]&gt;</REPORTEDPRIORITY>
     <EXTERNALRECID>NETCOOL</EXTERNALRECID>
     <EXTERNALSYSTEM_TICKETID>NCOMS/953</EXTERNALSYSTEM_TICKETID>
               <MULTIASSETLOCCI action="Add"><RECORDCLASS>
               <![CDATA[INCIDENT]]&gt;</RECORDCLASS><ASSETNUM><![CDATA[7200]></ASSETNUM>
               </MULTIASSETLOCCI>
       </TNCTDENT>
  </MXINCIDENTSet>
</CreateMXINCIDENT>
```

#### Sending data from ICD to the ObjectServer

When sending data from ICD to the ObjectServer, the gateway uses TAL scripting support. This provides functionality similar to that provided by rules files. To configure additional fields to be sent from ICD to the ObjectServer add the fields to the \$OMNIHOME/gates/tsrm/tsrm.script file using the following example. In this example the TicketStatus field is set based on the STATUS token value.

**Note :** The TicketStatus field is the only field enabled for bidirectional operation by default. For further information on extending the availability of ICD fields, see <u>"Adding an ICD field to the list of fields</u> retrieved by the gateway" on page 35.

#### **Table replication definition file**

The gateway replicates data between ObjectServer tables and the gateway target. The table replication definition file is used to define which tables and event types are monitored in Tivoli Netcool/OMNIbus and forwarded to the target that the gateway is configured to send data to.

You can specify the location of the table replication definition file using following generic Tivoli Netcool/ OMNIbus property.

#### Gate.Reader.TblReplicateDefFile

The default table replication definition file is in the following directory: OMNIHOME/gates/tsrm/tsrm.rdrwtr.tblrep.def

The default table replication definition file contains example commands. You should make a backup copy of the default file for future reference.

**Note :** You should use the REPLICATE command to replicate data from the primary tables (alerts.status, alerts.journal, alerts.details) and dynamic secondary tables (if required).

You can add one or more optional clauses to the REPLICATE command to further process the data during replication. The available commands are listed in the following syntax example. Use the optional clauses in the order in which they are listed in the syntax. For example, when using both the

FILTER WITH and AFTER IDUC DO clauses, the FILTER WITH clause must precede the AFTER IDUC DO clause.

```
REPLICATE ALL | (INSERTS, UPDATES, DELETES)
FROM TABLE sourcetable
USING MAP mapname
[FILTER WITH filter]
[INTO targettable]
[ORDER BY order, ...]
[AFTER IDUC DO afteriduc] ;
```

Table 11. Optional replication commands		
Command	Description	
FILTER WITH ' <i>filter</i> '	Filters the database rows selected for replication, where <i>filter</i> defines the filter that the gateway uses in the WHERE clause of the SQL_SELECT.	
	Filtering is positive by default, which means that only those events that match the filter definition are replicated. You can use a negative filter by putting an exclamation mark (!) before the equals sign (=) in the filter clause. For example, the following filter clause replicates all events whose severity is not 5: FILTER WITH 'Severity !=5'	
ORDER BY 'order'	Order results by the SQL SELECT ORDER BY clause used to get data. A potential use case might be to order by first occurrence, so that alerts are processed in chronological order, in which case the value specified for <i>order</i> would be 'FirstOccurrence'.	
AFTER IDUC DO 'afteriduc'	Updates replicated rows, where <i>afteriduc</i> specifies which field to update with what value. This uses the SQL UPDATE action to execute on rows retrieved by the SQL SELECT action used to get data, e.g. 'SentToCRM=1'.	

## **Advanced properties**

After configuring the basic property settings required to run the gateway, you might have to specify some advanced properties to suit your operating environment.

This section describes the property settings for the following functions:

- "Failback" on page 29
- <u>"Process Agent control" on page 30</u>
- <u>"Authentication" on page 30</u>
- <u>"Error handling" on page 30</u>
- "Creating service requests and incidents" on page 30
- "Polling ICD for unread tickets" on page 31
- "Passing data back to Netcool/OMINbus" on page 32
- "Encryption" on page 32
- "XML templates" on page 32

#### Failback

The failback function comes into operation when the gateway has lost its connection to the primary ObjectServer and has failed over to the backup ObjectServer. The backup ObjectServer is configured using the nco\_xigen utility.

To enable failback, set the **Gate.RdrWtr.FailbackEnabled** property to TRUE. When the reader has detected that it is now connected to a backup ObjectServer, it polls for the return of the primary ObjectServer with the frequency specified by the **Gate.RdrWtr.FailbackTimeout** property. When the primary ObjectServer is detected again, the reader automatically fails back to the primary ObjectServer.

## **Process Agent control**

The gateway can be run under Process Agent (PA) control.

The **Gate.PAAware** property specifies whether or not the gateway is PA-aware. The **Gate.PAAwareName** property specifies which PA is running the gateway.

**Note :** These properties are maintained automatically by the PA server and provide information only. Do not manually change the values of these properties.

#### **Authentication**

Use the Gate.UsePamAuth property to specify how the gateway authenticates users.

Either standard UNIX authentication or Pluggable Authentication Modules (PAM) authentication can be used with the Gateway for TSRM. By default, the gateway uses standard UNIX authentication. To use PAM authentication, set the **Gate.UsePamAuth** property to TRUE.

Note : To run the gateway in FIPS 140-2 mode, you must set the Gate.UsePamAuth property to TRUE.

#### **Error handling**

Error reporting is provided by the Netcool/OMNIbus Gateway Toolkit (NGTK) library. To specify that the NGTK library logs debug messages, set the **Gate.NGtkDebug** property to TRUE.

The debug files provide information about errors in the following areas:

- mapper syntax
- ObjectServer communication failure
- invalid column references in the mapper
- · attempts to update a field in the ObjectServer that does not exist
- ICD communication failure

If there is a communication failure with ICD, the gateway generates a GatewayWatch message. The gateway then retries the connection to ICD with the frequency specified by the

**Gate.TSRM.RetryInterval** property. If the gateway fails to connect to ICD after the number of times specified by the **Gate.TSRM.RetryInterval** property, the gateway shuts down.

For communication problems with ICD, the gateway sends a code back to the ObjectServer to identify the error. The code is sent to the ObjectServer field specified by the **Gate.TSRM.ErrorCodeFieldname** property. For example, if the gateway cannot send requests to ICD due to a communication error, the gateway sends error code 1 to the ObjectServer error code field. If there is an application error, the gateway sends error code 2 to the ObjectServer error code field.

#### **Creating service requests and incidents**

Within the ICD object structure, entities such as tickets and work logs are treated as separate objects and identified by name. Use the **Gate.TSRM.TicketObjectName** property to specify the object structure name that ICD uses for tickets and the **Gate.TSRM.WorkLogObjectName** property to specify the object structure name for work logs.

The gateway can support either ICD incidents or ICD service requests, but not both simultaneously. If you want to create both incidents and service requests, you must run two instances of the gateway, one for each type of ticket.

To configure the gateway to support either ICD incidents or ICD service requests you must specify the required class in the \$OMNIHOME/gates/tsrm/tsrm.map file. To support ICD incidents set class to 'INCIDENT' or to support service requests set class to 'SR'. For example;

CREATE MAPPING StatusMap ( 'CLASS' = 'SR', 'DESCRIPTION' = '@Node' + ":" + '@Summary' ON INSERT ONLY, 'REPORTEDBY' = 'NETCOOL' ON INSERT ONLY,

```
TO_TIME('@FirstOccurrence') ON INSERT ONLY,
       'REPORTDATE '
                           =
       REPORTEDPRIORITY' =
                                  Lookup('@Severity','SeverityTable'),
Lookup('@Severity', 'StatusTable'),
                     =
       'STATUS'
                                  '@TTNumber'
       'TTNumber'
CREATE MAPPING JournalMap
                            'SR',
         'CLASS'
                     =
         'Chrono'
                                    =
                                             '@Chrono'
                                              TO_TIME('@Chrono'),
'NETCOOL JOURNAL ENTRY'
         'CREATEDATE'
                                   =
         'DESCRIPTION'
                                    -
         'DESCRIPTION LONGDESCRIPTION'
                                                       =
                                                               TO STRING('@Text1') +
             TO_STRING('@Text2') + TO_STRING('@Text3')
);
```

Note : Class is set to 'INCIDENT' by default.

If you want the gateway to create and maintain ICD incidents, specify the following property values:

Property	Value
Gate.TSRM.URLList	http:// <targethost>/meaweb/os/MXINCIDENT</targethost>
Gate.TSRM.TicketObjectName	MXINCIDENT
Gate.TSRM.TicketObjectType	INCIDENT

If you want the gateway to create and maintain ICD service requests, specify the following property values:

Property	Value
Gate.TSRM.URLList	http:// <targethost>/meaweb/os/MXSR</targethost>
Gate.TSRM.TicketObjectName	MXSR
Gate.TSRM.TicketObjectType	SR

#### **Polling ICD for unread tickets**

The ICD reader thread polls ICD for all unread tickets and work log changes. This polling is based on a timestamp that the gateway maintains internally. Unread incident changes are processed as ObjectServer status table changes.

**Note :** To prevent miscommunication of ticket details, the time zones for the ObjectServer, the gateway, and the ICD server must be synchronized at all times.

The unread ticket changes are processed by the tsrm.script file supplied with the gateway. By default, any changes made to the STATUS field in ICD are transferred back to the corresponding field in the ObjectServer. You can transfer additional fields back to the ObjectServer by using the **Gate.TSRM.QueryRetrieveList** property. For example, if you want to indicate whether a Service

Level Agreement (SLA) has been applied to any of the tickets whose details are passed back to the ObjectServer, set the **Gate.TSRM.QueryRetrieveList** property to SLAAPPLIED.

**Note :** You must also add any additional fields that you want to send back to the ObjectServer to the XML structure for the service request or incident, the ObjectServer, and the tsrm.sql file supplied with the gateway. For details of how to do this, see <u>"Retrieving additional fields from ICD" on page 34</u>.

To improve the performance of the ICD reader, each time an incident or a work log object is created, the source attribute of the corresponding ICD item is populated with the text NETCOOL. You can use this as a filter during polling. You can specify an alternative text to identify tickets created from Netcool/OMNIbus using the **Gate.TSRM.SourceName** property.

If the poller cannot communicate with the ICD server, it generates the following GatewayWatch message:

TSRM Gateway is unable to connect to the TSRM Server

#### **Filtering repeated values**

The ObjectServer may send updates to the gateway which have unchanged event data since the last polling. This can put excessive load on the ICD server. To instruct the gateway to send only event data that has changed since the last polling, set the **Gate.TSRM.FilterRepeatedValues** property to true. This enables the gateway to filter messages based on the post mapped values.

To stop filtering on fields in which you are not interested in preventing repeat values, specify them using the **Gate.TSRM.FilterExclusionList** property. This property enables the gateway to stop filtering data which you do not need in terms of ObjectServer updates. For example, Date and Time fields usually fall into this category and you may need to know when the severity of an event has changed and what date it occurred but you may not need all event reoccurrences with the same severity.

## Removing reliance on the ICD host clock

The Gateway for TSRM uses the CHANGEDATE value to determine new ICD updates sent by ICD.

If the **Gate.TSRM.PollTimeIntervalWindow** property is set to a positive value, the gateway uses the following algorithm to query for updates on ICD:

CHANGEDATE > (current time using the gateway host clock) - (Gate.TSRM.PollTimeIntervalWindow)

This removes the reliance on the ICD host clock.

Note : The clocks on the gateway server and the ICD server need to be in synch.

For best performance the **Gate.TSRM.PollTimeIntervalWindow** value should be twice the value of set for the **Gate.TSRM.PollTime** property.

**Note :** The polling for events on ICD will still occur with the frequency specified by the **Gate.TSRM.PollTime** property.

## Passing data back to Netcool/OMINbus

To enable the gateway to query and maintain ICD tickets that originated in Netcool/OMNIbus, and to pass data back to Netcool/OMNIbus, you must specify which fields in the ObjectServer alerts.status table hold the corresponding details from ICD.

You do this using the following properties:

- Gate.TAL.TTNumberFieldName: this field is required to hold the trouble ticket number.
- **Gate.TAL.TTStateFieldName**: this field is required to hold the integer that indicates the state of the trouble ticket.
- Gate.TAL.TTJournalChronoFieldName: this field is required to prevent duplicates during failback.

#### Encryption

You can use encryption algorithms (for example, AES or DES) to secure the entries made in the properties file.

Use the **ConfigCryptoAlg** property to specify the encryption method and the **ConfigKeyFile** property to specify the encryption key data file.

## **XML** templates

The gateway writes Netcool/OMINbus data to ICD tickets by using predefined XML templates.

The templates are supplied with the gateway as individual text files and are specified using the following properties:

- Gate.TSRM.CreateTicketTemplate: the gateway uses this template to create tickets in ICD.
- Gate.TSRM.UpdateTicketTemplate: the gateway uses this template to update ICD ticket details.
- Gate.TSRM.JournalTemplate: the gateway uses this template to create work logs in ICD.

• Gate.TSRM.QueryTicketTemplate: the gateway uses this template to query ticket details in ICD.

# **Configuring SSL connections**

If you are using an HTTPS/SSL connection between the gateway and ICD, you must create a truststore to store the ICD digital certificate and point the gateway to the location of the truststore.

You can generate the truststore file using the Java keytool utility. The keytool utility is located in the following directory:

\$OMNIHOME/platform/arch/jre\_directory/jre/bin/

Where *arch* is the operating system you are running and *jre\_directory* is the installation directory of your Java Runtime Environment (JRE).

Use the following steps to enable SSL connections between the gateway and ICD:

1. Obtain the ICD certificate files.

**Note :** You must obtain the CA root certificate and any intermediate CA certificates that your ICD configuration uses.

- 2. Use the Java keytool utility to create a truststore and password.
- 3. Add the ICD SSL certificate to the truststore.

For example, run the following keytool command from the \$NCHOME/platform/arch/ jre\_version/bin directory to import the certificate and create the keystore:

keytool -import -trustcacerts -alias CAROOTCERT -file PATH CA.cer -keystore
\$NCHOME/etc/security/tsrmcerts

Note : You will be prompted to create a truststore password.

4. Update the Gate.TSRM.URLList property to include the HTTPS MEA location.

For example: https://targethost/meaweb/os/MXINCIDENT

**Note :** The *targethost* variable must match the hostname specified in the ICD server certificate.

- 5. Update the Gate.Java.Arguments property to include the following arguments:
  - -Djavax.net.ssl.trustStore=truststore\_path

Where *truststore\_path* is the full path and name of the truststore file.

-Djavax.net.ssl.trustStorePassword=truststore\_password

For example:

```
Gate.Java.Arguments : '-Djavax.net.ssl.trustStore=/opt/IBM/tivoli/netcool/
platform/arch/jre_1.6.7/jre/lib/security/tsrmcerts -
Djavax.net.ssl.trustStorePassword=changeit'
```

## **Configuring ICD**

After installing the gateway, you must configure ICD to run with the gateway.

The following sections explain how to configure ICD to run with the gateway:

- "Configuring ICD to receive journal entries from Netcool/OMNIbus events" on page 33
- "Retrieving additional fields from ICD" on page 34
- "Configuring the ICD event summary field" on page 36

#### Configuring ICD to receive journal entries from Netcool/OMNIbus events

Within ICD, entities such as incidents and work logs are represented using objects. These objects also have XML representations. Before using the gateway, you must configure ICD to support the passing of

journal entry details from Netcool/OMNIbus to ICD. These entries then get created as work logs within ICD.

To configure ICD to receive journal entries from Netcool/OMNIbus events, you must add the structure for work logs to that of incidents and service requests within the ICD object structure. You will be able to create journal entries on Netcool/OMNIbus events and use the Gateway for TSRM to pass those details to their corresponding ICD tickets as work logs.

To add the appropriate XML structure required to add work logs to the ICD object structure, use the following steps:

- 1. Start the ICD Web application interface.
- 2. Select Go To > Integration > Object Structures.
- 3. In the filter field of the **Object Structure** column, enter MXINCIDENT and press **Enter**.
- 4. Open the MXINCIDENT object structure.

**Note :** To add the work order object to the structure for service requests instead of incidents, search for, select, and open MXSR instead of MXINCIDENT. Integration Application is responsible for all ICD interfaces with external applications. In this context, the external application is the Gateway for TSRM.

- 5. Select the **New Row** button at the bottom of the **Source Object** panel.
- 6. Select **INCIDENT** from the Object drop-down field.

Note : To add the work order object to the structure for service requests, select SR.

- 7. Add a second new row to the **Source Object** panel.
- 8. Make the following selections for the second row:
  - a. From the Source Object drop-down field, select WORKLOG.
  - b. From the Parent Object drop-down field, select INCIDENT.

Note : To add the work order object to the structure for service requests, select SR.

- c. Select MODIFYWORKLOG from the Relationship drop-down field.
- 9. Select Select Action > Exclude/Include Fields.

The interface displays the **Exclude Include Fields** dialog box.

- 10. In the **Source Objects** list, select the **WORKLOG** object.
- 11. Select the **Non-Persistent Fields** tab.
- 12. Select the **DESCRIPTION\_LONGDESCRIPTION** check box.
- 13. Select **OK** to close the dialog box.
- 14. Select Select Action > Generate Schema/View XML.
- 15. Select **OK** to generate the XML schema required within the ICD object structure.

ICD displays the **View XML** dialog box.

16. Select **OK** to close the window.

## **Retrieving additional fields from ICD**

By default, the gateway only retrieves the STATUS field from ICD tickets.

If you want to retrieve an additional field from ICD to send back to Netcool/OMNIbus, you must use the following steps:

- 1. Add the field to the XML structure for the ICD incident or service request.
- 2. Add the field to the ObjectServer.
- 3. Add the ICD field to the list of fields retrieved by the gateway.

#### Adding fields to the XML structure for incidents and service requests

You can add any of the ICD ticket fields to the details that are retrieved and sent back to Netcool/ OMNIbus. For details of which fields are available for selection, see the IBM Control Desk documentation.

The following example shows how to add the appropriate XML code for retrieving the SLAAPPLIED field to the XML structure for an ICD incident:

- 1. Start ICD.
- 2. Select Go To > Integration > Object Structures .
- 3. In the filter field of the **Object Structure** column, enter MXINCIDENT and press **Enter**.
- 4. Open the **MXINCIDENT** object structure.

**Note :** To add an ICD field to service request details retrieved and sent back to Netcool/OMNIbus, search for, select, and open MXSR instead of MXINCIDENT.

- 5. Select the **Object Structure** tab.
- 6. Select the Exclude/Include Fields action.

ICD displays the **Exclude/Include Fields** dialog box.

- 7. Select the Non-Persistent Fields tab.
- 8. Select the Include check box beside the SLAAPPLIED entry.
- 9. Click **OK**.
- 10. To confirm that the XML code for the SLAAPPLIED field has been added correctly, select the **Generate Schema/View XML** action from the **Object Structure** panel.

ICD displays the View XML dialog box.

11. In the left-hand side of the panel, scroll through the XML and locate the following XML (highlighted in Figure 12):

<max:SLAAPPLIED changed="false">false</max:SLAAPPLIED>

12. Select **OK** to close the dialog box.

#### Adding a field to the ObjectServer

To add a field to the ObjectServer, run the following command using the ObjectServer SQL tool:

```
alter table alerts.status add column FieldName typ; go
```

where FieldName is the name of the field that you are adding and typ indicates the data type.

For example, run the following command to add the SLAApplied field to the ObjectServer:

```
alter table alerts.status add column SLAApplied int; go
```

#### Adding an ICD field to the list of fields retrieved by the gateway

To add an ICD field to the list of fields retrieved by the gateway, set the **Gate.TSRM.QueryRetrieveList** property to SLAAPPLIED, where SLAAPLIED is the ICD ticket field that you want to add.

You must also add the extra ICD field to the following SQL file that is supplied with the gateway:

\$OMNIHOME/gates/tsrm/tsrm.script

The following example shows the added SLAAPLIED field in bold type:

CASE\_OP\_TYPE STATUS\_UPDATE: update alerts.status via '\$Identifier' set **SLAAPLIED='\$SLAPPLIED'** where ServerName='\$ServerName' and ServerSerial=\$ServerSerial;

```
BREAK;
END;
```

#### Checking that the new field gets sent back to Netcool/OMNIbus

To check that the SLAAPPLIED field gets sent back to Netcool/OMNIbus, use the following steps:

- 1. Run the gateway.
- 2. Start ICD.
- 3. Open an incident created by the gateway.
- 4. Apply an SLA to the incident by selecting Select Action > Apply SLA
- 5. Check the Netcool/OMNIbus Event List to verify that the SLA was applied.

### Configuring the ICD event summary field

To enable ICD to process event summary data of more than 100 characters, you must configure the description attributes of the TICKET, WORKVIEW, and WORKORDER objects.

The Netcool/OMNIbus event data that is passed to ICD is composed from data stored in the Node and Summary columns of the ObjectServer alerts.status table. By default, the alerts.status table limits the number of Node characters to 64 and Summary characters to 255. When the gateway concatenates the data in these two columns, the total character count sent to ICD is 320 (64 + 255 + 1).

The default setting of the ICD summary field for incidents or service requests is 100 characters. Because the gateway might be passing on summary data of more than 100 characters, this default setting can cause problems with the creation of incidents or service requests in ICD.

When this happens, you can expect to see errors similar to the following being logged in the gateway log file (\$OMNIHOME/log/GAT\_TSRM.log):

2010-09-23T18:24:17: Error: E-GJA-000-000: [ngjava]: TSRMRdrWtr: Error 500: nested exception is: psdi.util.MXApplicationException: BMXAA4049E -The value specified <event data being sent to TSRM> exceeds the maximum field length.

To avoid possible problems with the creation of summary data, you must change the description attributes of the TICKET, WORKVIEW, and WORKORDER objects in ICD. This will enable ICD to process summary field lengths greater than 100 characters.

#### Setting object description attributes

To enable ICD to process summary field lengths of up to 320 characters, use the following steps:

- 1. Start the ICD Web application interface.
- 2. Select Go To > System Configuration > Platform Configuration > Database Configuration.
- 3. For each of the TICKET, WORKVIEW, and WORKORDER objects, use the following steps:
  - a. Search for the object.
  - b. Select the object.
  - c. On the Attributes tab, choose the DESCRIPTION attribute.
  - d. In the attribute details panel, set the character length to 320.
  - e. Save your changes.

#### Activating the new description character lengths

To activate the new description character lengths, use the following steps:

1. Select Go To > System Configuration > Platform Configuration > Database Configuration.

- 2. Select Select Action > Manage Admin Mode.
- 3. Select Turn Admin Mode On.
- 4. Refresh the status until the Admin Mode On message appears.
- 5. Select Select Action > Apply Configuration Changes.
- 6. In the Admin Mode window, select Close.
- 7. Select the Current Backup checkbox.
- 8. Select Start Configuring the Database.
- 9. Refresh the status until configuration is complete and then select **OK**.
- 10. Select Select Action > Manage Admin Mode.
- 11. Select Turn Admin Mode Off.
- 12. When the admin mode turns off, select **Close**.

## **Configuring ICD for improved incident reporting performance**

The default object structure used in ICD sends larger than necessary message to the gateway. To enable ICD to send smaller MXINCIDENT messages to the gateway use the following instructions.

The Gateway for TSRM uses direct HTTP posts to the object structure servelet in ICD to log Create, Update and Query incidents.

The Query response will be larger than necessary if you use the standard MXINCIDENT object structure and configuration from <u>"Configuring ICD to receive journal entries from Netcool/OMNIbus events" on page 33</u> because you add the WORKLOG object to the MXINCIDENT object structure to enable the journaling feature to work.

To enable the Gateway for TSRM to receive the information it requires only, you must create a new object structure and the configure the gateway to send information to the HTTP servelet this creates.

The gateway requires the following attributes to be sent to or received from ICD on the INCIDENT object:

- CHANGEDATE
- CLASS
- DESCRIPTION
- EXTERNALSYSTEM
- EXTERNALSYSTEM\_TICKETID
- REPORTEDBY
- REPORTDATE
- REPORTEDPRIORITY
- STATUS

The gateway requires the following attributes for the create incident:

- CLASS
- CREATEDATE
- DESCRIPTION
- DESCRIPTION\_LONGDESCRIPTION

To create the new object structure use the following example:

- 1. Login to ICD and select: Integrations > Object Structures > New Object Structure
- 2. Specify the following fields in the new object structure:
  - Object Structure: OMNIINCIDENT
  - Consumed By: INTEGRATION
- 3. Select New Row and specify Object: INCIDENT

- 4. Select **New Row** and specify the following:
  - Object: WORKLOG
  - Parent Object: INCIDENT
  - Relationship: MODIFYWORKLOG
- 5. Click Save
- 6. Now you exclude the attributes you do not want to use. Click Select Action > Exclude/Include Fields
- 7. Select the Incident Object and exclude all attributes except:
  - CHANGEDATE
  - CLASS
  - DESCRIPTION
  - EXTERNALSYSTEM
  - EXTERNALSYSTEM\_TICKETID
  - REPORTEDBY
  - REPORTDATE
  - REPORTEDPRIORITY
  - STATUS
- 8. Select the WORKLOG object and exclude all attributes except:
  - CLASS
  - CREATEDATE
  - DESCRIPTION
- 9. Select the **Non-Persistent Field tab** with WORKLOG still highlighted and include the DESCRIPTION\_LONGDESCRIPTION attribute.
- 10. Click OK then Save
- 11. Specify the following properties in the gateway properties file to configure the TSRM gateway to use this new servelet:
  - Gate.TSRM.URLList: http://<servername>/meaweb/os/OMNIINCIDENT'
  - Gate.TSRM.TicketObjectName: 'OMNIINCIDENT'
- 12. Restart the gateway.

#### Fine tuning TSRM gateway event processing

In a high transaction volume environment the TSRM gateway may experience memory growth in a consistent way.

Some of the following solutions may enable ICD to process events faster in the case of slow processing performance.

• Increase the number of MEA servers/JVMS in ICD to handle multiples events as one time.

Note : You may need to contact your ICD administrator.

- Increase the **ThreadPoolSz** value specified in the gateway properties file to increase the number of events to send to ICD at one time.
- Increasing the resource allocation for the DB2 server as it can speed up event processing and resolve performance issues. If slow processing is occurring in ICD DB2 this can cause the gateway cache to increase in order to store the incoming event. To identify this problem the user can look at the time the operations are taking in ICD for one particular event. The slow processing on the DB2 side could be due to disk IO issues caused by incorrect system configuration such as low memory capacity on DB2.

# **Configuring Netcool/OMNIbus**

After installing the gateway, you must configure Netcool/OMNIbus to run with the gateway.

The following sections explain how to configure Netcool/OMNIbus to run with the gateway:

- "Configuring log ticketing functions in Netcool/OMNIbus" on page 39
- "Configuring the Netcool/OMNIbus Event List to use the launch-in-context tool" on page 39

## **Configuring log ticketing functions in Netcool/OMNIbus**

Add log ticketing functions to the Netcool/OMNIbus Event List.

You must add log ticketing functions to the Netcool/OMNIbus Event List using the following steps:

- 1. Select Netcool Suite > System Utilities > Server Editor.
- 2. Configure the Server Editor as required.
- 3. Select Netcool Suite > Administrator.
- 4. Highlight the ObjectServer.
- 5. Open the ObjectServer by double-clicking it.
- 6. Log in to the ObjectServer.
- 7. Select Tools > Item > Add tool.
- 8. Enter Log Ticket.
- 9. Select the **Enabled** and **Execute** checkboxes for each selected row.
- 10. In the SQL window, enter the following command from tools.sql:

```
update alerts.status set LogTicket=1 where Serial in
($selected_rows.Serial); flush iduc;
```

Note : The tools.sql file is supplied with the gateway installation package.

To add the tool to a menu, use the following steps:

- 1. Select Menus > AlertsMenu > <expand> > &Tools.
- 2. Right-click on **&Tools**.
- 3. Select Add Menu Item.
- 4. Enter the following field values:

Field	Value
Menu Item	Type Tool
Tool	Log Ticket
Title	Log Ticket

5. Select OK.

# Configuring the Netcool/OMNIbus Event List to use the launch-in-context tool

The launch-in-context tool allows you to access IBMControl Desk from the Netcool/OMNIbus Event List to view details of ICD tickets that correspond to Netcool events.

To use the launch-in-context tool, you must add it to the list of available Netcool/OMNIbus tools using the Tool Creation option within Netcool/OMNIbus Web GUI. You must then add the tool to the list of tools that appear on the pop-up menu in the Netcool/OMNIbus Event List using the Menu Configuration option within Netcool/OMNIbus Web GUI.

This section describes how to perform both these tasks. For full details of how to use Netcool/OMNIbus Web GUI, see the *IBM Tivoli Netcool/OMNIbus Web GUI Administration and User's Guide* (SC14-7606-00).

#### Adding the launch-in-context tool to the list of those available to the Netcool/ OMNIbus event list

To add the launch-in-context tool to the list of those available, use the following steps:

- 1. Log on to Netcool/OMNIbus Web GUI as an administrator.
- 2. Select Administration > Event Management Tools > Tool Creation
- 3. Set the Name field to ViewTicket.
- 4. From the Type drop-down field, select New > CGI/URL
- 5. In the **URL** field, enter the following URL:

```
http://tsrmhost/maximo/ui/?event=loadapp&value=incident
&additionalevent=useqbe&additionaleventvalue=ticketid={$selected_rows.TTNumb
er}
```

where tsrmhost is the ICD server.

- 6. Select the Execute for each selected row check box.
- 7. Select Save.

ViewTicket now appears in the Tool Creation list of tools.

# Adding the tool to the list that appear on the pop-up menu in the Netcool/OMNIbus event list

To add the tool to the list that appears on the pop-up menu in the Netcool/OMNIbus event list, use the following steps:

- 1. Log on to Netcool/OMNIbus Web GUI as an administrator.
- 2. Select Administration > Event Management Tools > Menu Configuration.
- 3. Select Alerts > Modify

Netcool/OMNIbus Web GUI displays the Menus Editor dialog box.

- 4. Right click on the selected event and select ViewTicket from available items.
- 5. Select the right-arrow button to add ViewTicket to the **Current Items** list.
- 6. Select **Save**.

# Using the launch-in-context tool to access IBM Control Desk from within the Netcool/OMNIbus event list

- 1. Log on to Netcool/OMNIbus Web GUI as an administrator.
- 2. Select Availability > Events > Active Event List (AEL).
- 3. From the list of active events, select an event for which you have to create a corresponding ticket in ICD (one for which there is an identifier listed in the **TTNumber** column).
- 4. Select Alerts > ViewTicket.
- 5. Enter your user name and password.
- 6. Select Sign In.
- 7. Select the ticket number in ICD to view the incident or service request.

ICD displays the details of the ICD ticket that corresponds to the Netcool/OMNIbus Event.

# **Running the gateway**

This topic describes how to run the gateway. On Windows operating systems, you can run the gateway as a service.

To start the gateway on UNIX and Linux operating systems, run the following command:

\$OMNIHOME/bin/nco\_g\_tsrm -propsfile \$OMNIHOME/etc/GAT\_TSRM.props

To start the gateway as a process on Windows operating systems, run the following command:

%OMNIHOME%\bin\nco\_g\_tsrm.exe -propsfile %OMNIHOME%\etc\GAT\_TSRM.props

To start the gateway as a service on Windows operating systems, use the following steps:

- 1. Run the ObjectServer as a service.
- 2. Register the gateway with the Service Control Manager.
- 3. If the gateway and the ObjectServer are running on the same host, run the following command: %OMNIHOME%\bin\nco\_g\_tsrm.exe -install -depend NCOObjectServer.
- 4. If the gateway and the ObjectServer are running on different hosts, run the following command: %OMNIHOME%\bin\nco\_g\_tsrm.exe -install.
- 5. Start the gateway using the Microsoft Services Management Console.

# **FIPS mode and encryption**

This gateway complies with Federal Information Processing Standard 140-2 (FIPS 140-2). It can be run in FIPS mode on any currently supported version of Tivoli Netcool/OMNIbus.

You can use encryption algorithms to secure string value entries made in the properties file, including passwords. You must use the generic Tivoli Netcool/OMNIbus **ConfigCryptoAlg** property to specify the encryption method and the generic Tivoli Netcool/OMNIbus **ConfigKeyFile** property to specify the encryption key file, amongst a number of other required settings.

For more information about running the gateway in FIPS mode, and encrypting properties and passwords, see *Running the ObjectServer in secure mode*, *Running the proxy server in secure mode*, and *Encrypting plain text passwords in routing definitions* in the *IBM Tivoli Netcool/OMNIbus Administration Guide*.

Also see, Configuring FIPS 140-2 support for the server components in the IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide.

Also see SSL and FIPS 140-2 support in the IBM Tivoli Netcool/OMNIbus Event Integration Facility Reference.

Also see Appendix C. WAAPI security in the IBM Tivoli Netcool/OMNIbus Web GUI Administration API (WAAPI) User's Guide.

**Note :** If you run the gateway in FIPS mode, you must either use no encryption, or if you do use encryption, you must use nco\_aes\_crypt with the cipher (-c) option AES\_FIPS. The cipher option used here must match the option specified by the **ConfigCryptoAlg** property. For example:

\$NCHOME/omnibus/bin/nco\_aes\_crypt -c AES\_FIPS -k key\_file string\_value

# **Testing Netcool/OMNIbus communication with ICD**

This topic describes how to verify that Netcool/OMNIbus is successfully communicating with ICD.

**Note :** The gateway does not support resynchronization. That means that you can only submit a request for a ticket to be created while the gateway is running. If you submit the request while the gateway is not running, you will have to resubmit the request after you have restarted the gateway.

To confirm that Netcool/OMNIbus is successfully communicating with ICD, use the following steps:

1. Start the gateway.

2. Start the Netcool/OMNIbus Events List.

- 3. Right-click on an event.
- 4. Set the LogTicket field to 1.

When the incident has been created within ICD, a TTNumber is assigned (this is the incident number within ICD) and the **TicketStatus** field in the Event List is set to **NEW**.

5. To verify that a change to the incident's status is routed back to the Event List, change the status of the ticket in ICD to In Progress using the **Change Status** dialog box.

When the gateway next polls ICD, the updated status will be passed back to the ObjectServer and will appear in the Event List as **INPROG**.

6. To verify that journal entries you create for events linked to ICD incidents result in the creation of corresponding work logs in ICD, create a journal entry as shown below.

When you check the incident in ICD , the work log is visible in the Work Logs tab.

## **Error messages**

Error messages provide information about problems that have occurred during the operation of the gateway. You can use the information that they contain to resolve such problems.

Table 12. Error messages Error Description Action Failed to get a handle The gateway could not initialize Verify that the properties file has the gateway properties. correct permissions and check the to the properties values set in the properties file. manager. Invalid close on delete The Gate.TSRM.CloseOn Ensure that the property setting Name: **DeleteList** property is Gate.TSRM.CloseOn name Value: value incorrectly specified. **DeleteList** property is specified using comma-delimited namevalue pairs in the following format: Name1='Value1'. ....NameN='ValueN'. For example: CLASS='INCIDENT' STATUS='RESOLVED' Unable to parse the The Gate.TSRM.CloseOn Ensure that the Close on delete list DeleteList property is Gate.TSRM.CloseOn incorrectly specified. **DeleteList** property is specified using comma-delimited namevalue pairs in the following format: Name1='Value1', ... ,NameN='ValueN'. For example: CLASS='INCIDENT' STATUS='RESOLVED'

The following table describes the error messages that the Gateway for TSRM generates.

Table 12. Error messages (continued)		
Error	Description	Action
Error processing northbound script	The gateway is unable to process ticket updates from ICD to the	Verify that the gateway can communicate with ICD.
error_message	ObjectServer.	Communication between the gateway and ICD can be tested using the following steps:
		1. Stop the gateway.
		<ol> <li>Specify an empty value ("") for the Gate.TAL.NBScript FileName property.</li> </ol>
		This (in effect) puts the gateway in unidirectional mode.
		3. Restart the gateway.
		4. Create a single ticket in ICD and check if there is communication with the gateway.
EXTERNALSYSTEM_TICKETID does not contain the netcool alert keys !!	The response to an ICD poll does not contain a valid value for ServerName and ServerSerial (which together provide the key for the alert in the ObjectServer). This means that the gateway is getting an update for a ticket that it did not create.	Verify that the EXTERNALSYSTEM_TICKETID XML Response element contains a value in the format ServerName/ServerSerial, where ServerName is a string and ServerSerial is an integer.
Unable to create TSRM Notification processor: exception_message	The gateway is unable to create a thread to process updates from ICD to the ObjectServer.	The <i>exception_message</i> included in the error message indicates the possible cause of the error.
		The most common of several possible causes is that the <b>Gate.TAL.NBScriptFileName</b> property is not pointing to a valid file. The default value of the property is \$OMNIHOME/gates/ tsrm/tsrm.script.
		<b>Note :</b> Specifying an empty value for this property causes the gateway to operate in unidirectional mode.

Table 12. Error messages (continued)		
Error	Description	Action
Failed to carry out TSRM operation for row Server Serial Num: server_serial_number Server Name: server_name MxConnHttpException: exception_message Error is: error_message Error code for operation is error-code Failed to feedback the error code: exception_message	A gateway-ICD operation has failed. The server_name and server_serial_number indicate which alert the error applies to. Error code 1 indicates that the error is a connectivity issue. Errors with error code 1 are transient and the operation can be retried. Error code 2 indicates an error where the operation cannot be repeated (such as invalid values for fields).	The exception_message included in the error message gives the description of the error. If the error code indicates a type 2 error, examine the tsrm.map file and verify that the correct values are being sent to the ICD server. It might also be the case that some errors are transient but do not result in a network exception being thrown (such as the shutting down of the ICD application in Websphere). This can be dealt with by using the <b>Gate.TSRM.NonApplication</b> <b>ErrorTitles</b> property to specify a comma-delimited list of application error titles (sent by ICD) that the gateway should interpret as connection errors. Errors in this list are treated as type 1 errors and the operation is retried.
There is not enough memory available	The operating system could not allocate enough memory to run the embedded JVM for the gateway.	Use the <b>Gate</b> .Java.Arguments property to allocate sufficient heap memory for the embedded JVM. For example, specifying a value of -Xmx1024m for this property will allocate 1GB of heap memory. <b>Note</b> : For AIX systems, the number of segments that a process can use for data is controlled by the LDR_CNTRL environment variable. For example, the following statement defines one additional data segment: export LDR_CNTRL =MAXDATA=0x10000000

# **GatewayWatch messages**

GatewayWatch messages provide information about how the gateway is running.

The following table describes the GatewayWatch messages that the Gateway for TSRM generates.

Table 13. GatewayWatch messages		
GatewayWatch message	Description	Cause
TSRM Gateway is unable to connect to the TSRM Server	The gateway poller is unable to communicate with the ICD server.	The gateway attempted to connect to the ICD server, but the server was unavailable. This was probably due to a network connectivity problem.

## **Known issues**

This section describes some known issues with the TSRM gateway.

#### CTRL-C shutdown gateway error message

When using the CTRL-C command to shutdown the gateway a error log similar to the one in the following example can sometimes occur. You can safely ignore this shutdown error.

#### FileNotFoundException: derby.logerror

This error message occurs when you do not have write access permissions to the folder where you are running the TSRM gateway.

To fix this issue change the permissions of the ICD directory to give the user running the gateway write access permissions or launch the gateway from a directory with write access permissions already granted.

46 IBM Tivoli Netcool/OMNIbus Gateway for TSRM: Reference Guide

# **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 wish 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:

<sup>©</sup> (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. <sup>©</sup> 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<sup>®</sup>, 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.

50 IBM Tivoli Netcool/OMNIbus Gateway for TSRM: Reference Guide



Part Number:

SC27-2703-10



(1P) P/N: