IBM® Tivoli® Netcool/OMNIbus Gateway for Amdocs CRM
2.0

Reference Guide
November 8, 2013

SC22-5410-02

Notice

Before using this information and the product it supports, read the information in Appendix A, “Notices and Trademarks,” on page 21.

Contents

About this guide

The following sections contain important information about using this guide.

Document control page

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

The Gateway for Amdocs CRM 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
SC22-5410-00	December 2, 2011	First IBM publication.
SC22-5410-01	November 30, 2012	Guide updated for Netcool/OMNIbus V7.4 release.
SC22-5410-02	November 8, 2013	"Summary" on page 1 "Startup command file" on page 6 added. "Table replication definition file" on page 7 added. "FIPS mode and encryption" on page 18 added.

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 *arch***
AIX® systems	`aix5`
Red Hat Linux® and SUSE systems	`linux2x86`
Linux for System z®	`linux2s390`
Solaris systems	`solaris2`
Windows systems	`win32`

OMNIHOME location

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

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

Chapter 1. Gateway for Amdocs CRM

This guide describes the integration of IBM Tivoli Netcool/OMNIbus and Amdocs CRM through the Gateway for Amdocs CRM. It is assumed that the reader has administrative experience with Netcool/OMNIbus, as well as a basic understanding of the Amdocs CBO API.

This guide contains the following sections:

Summary

The following table provides a summary of the Gateway for Amdocs CRM.

Table 3. Summary	
Gateway target	Amdocs CRM version 7.5 **Note :** For details of the versions of Amdocs that this gateway supports, see "Compatibility" on page 2.
Gateway executable name	`nco_g_amdocscrm`
Package Version	2.0
Gateway supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website:http://www-01.ibm.com/support/docview.wss?uid=swg21572340
Connection Interface	Amdocs CBO API

Configuration files	`$OMNIHOME/gates/amdocscrm/amdocscrm.map`
	`$OMNIHOME/gates/amdocscrm/AMDOCSCRM.props`
	`$OMNIHOME/gates/amdocscrm/amdocscrm.rdrwtr.tblrep.def`
	The following versions of the dependencies are the minimum requirements. The latest GA versions of the dependencies should be used.
	`$OMNIHOME/gates/amdocscrm/amdocscrm.sql`
	`$OMNIHOME/gates/amdocscrm/amdocscrm.startup.cmd`
	`$OMNIHOME/gates/amdocscrm/amdocscrm.script`
	`$OMNIHOME/gates/amdocscrm/httpTransport.properties`

Table 3. Summary (continued)	

Requirements	A currently supported version of Tivoli Netcool/OMNIbus
	Amdocs Web Application 7.5 must be installed on the Netcool server and the `TestCboInstallCpp` client must be configured to connect to the Amdocs server.
	Oracle 10.2 must be installed on the Netcool server.
	`OMNIbus-`*`arch`*`-gateway-libngtktk-2_x`
	`OMNIbus-`*`arch`*`-gateway-libngjava-3_x`
	`Omnibus-`*`arch`*`-common-transportmodule-3_x`
	`OMNIbus-`*`arch`*`-gateway-libtal-4_x`

Compatibility

This sections describes the platforms and databases with which the Gateway for Amdocs CRM is compatible.

Solaris

The following section mentions the availability of Gateway for Amdocs CRM on the Solaris platform.

The Solaris Gateway for Amdocs CRM is available for the following versions of Amdocs:

- Oracle 10g is supported with Amdocs CRM 7.5

Gateway overview

Amdocs CRM consists of a suite of applications designed to support a range of different business functions related to customer support.

The Gateway for Amdocs CRM will implement the integration between Tivoli Netcool/OMNIbus and Amdocs CRM. The Gateway for Amdocs CRM uses TAL to provide a uniform ticketing gateway framework and to integrate with Amdocs CRM the gateway uses the Amdocs CBO API.

The gateway provides the following functions:

- The Gateway for Amdocs CRM creates and updates Amdocs CRM cases based on alert data.
- The Gateway for Amdocs CRM changes the ticket status of the created Amdocs CRM cases based on values specified in the `amdocscrm.rdrwtr.tblrep.def` file.
- The Gateway for Amdocs CRM sends created cases to queues based on values specified in the `amdocscrm.map` file.
- The Gateway for Amdocs CRM provides bidirectional support to enable the Amdocs server to forward updated status requests to the ObjectServer.

Gateway alert classes

The gateway uses three Amdocs CRM gateway classes to manage Amdocs CRM alert cases:

- The `AmdocsCRMRdrWtr` class provides the implementation of the TALRdrWtr that supports the callback methods for creating and updating cases as well as creating the southbound notification processor. It uses a private class called CreateSession which is used to create the initial shared Amdocs CBO API session
- The `AmdocsCRMCbo` class provides the Amdocs CBO API helper methods that allow error handling and assigning Amdocs CRM alert case members.

- The `AmdocsNotificationProcessor` class implements the southbound processing by using the transport module in order to receive HTTP notification data from Amdocs CRM. This data is then tokenized and processed by the `amdocscrm.script` file.

Gateway operation

The gateway passes information between the Amdocs CRM reader and writer modules and the ObjectServer. New alerts are passed through the gateway to form Amdocs CRM cases. The gateway also feeds back the Amdocs CRM case ID of new cases to the ObjectServer so that alerts can be linked with associated cases.

Whenever an alert is updated or deleted in the ObjectServer, the changes are mapped through the gateway to modify the appropriate Amdocs CRM case. Likewise, if the alert journal is inserted into the ObjectServer, the changes are mapped through the gateway and reflected in the log notes of the associated Amdocs case.

If the gateway is bidirectional, changes to any case that was originally generated by Netcool/OMNIbus are passed through the gateway and reflected in the associated alert.

Note : If an alert is deleted in Netcool/OMNIbus, the associated case can be closed in Amdocs CRM by setting the **`Gate.AmdocsCrm.CloseOnDeleteList`** property.

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 the gateway on Tivoli Netcool/OMNIbus V7.4.0

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

You can install the gateway using any of the following:

- "The installation wizard" on page 4
- A text-based installer ("Console mode" on page 4)
- Settings predefined in a text file ("Silent mode" on page 4)

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 Netcool packages

The following Netcool packages must be installed in the order shown:

1. `Omnibus-`*`arch`*`-common-transportmodule-3_x`
2. `OMNIbus-`*`arch`*`-gateway-libngjava-3_x`
3. `OMNIbus-`*`arch`*`-gateway-libngtktk-2_x`
4. `OMNIbus-`*`arch`*`-gateway-libtal-4_x`
5. `OMNIbus-`*`arch`*`-gateway-nco-g-amdocscrm-2_x`

Note : If a newer version for any of these files is available it may be used instead of the older file mentioned in this list.

Configuring TAL to run on the ObjectServer

To run the TAL automations on the ObjectServer use the following commands:

cd `$OMNIHOME/bin`

`nco_sql –U <`*`omnibususer`*`> -P <`*`omnibuspassword`*`> <../gates/tal/tal_automations.sql>`

`nco_sql –U <omnibususer> -P <omnibuspassword> <../gates/amdocscrm/amdocscrm.sql`

Where *omnibususer* is the name of the user authorized to run Netcool/OMNIbus.

Where *omnibuspassword* is the password associated with this user.

Configuring the Gateway for Amdocs CRM

Use the following steps to configure the Gateway for Amdocs CRM.

1. Set values for the CLARIFY_HOME and ORACLE_INSTALL_DIR variables in the environment variables file:

   `$OMNIHOME/platform/solaris2/bin/nco_g_amdocscrm.env`

   For example:

   - `CLARIFY_HOME="/space/AmdocsCRM7.5/AmdocsCRMApplication"`
   - `ORACLE_INSTALL_DIR="/space/oracle_instant_client-10.2.0.4"`
2. Run `$OMNIHOME/bin/nco_xigen` and add AMDOCSCRM to the Netcool/OMNIbus server editor list.
3. Copy the Amdocs CRM properties file `$OMNIHOME/gates/amdocscrm/AMDOCSCRM.props` to `$OMNIHOME/etc` and set the **`Gate.AmdocsCrm.HttpServerPropsFile`** property so that it specifies the location of the `httpTransport.properties` file, for example:

   `'$OMNIHOME/gates/amdocscrm/httpTransport.properties'`
4. Set values for the **`ServerPort`**, **`Gate.AmdocsCrm.UserName`** and **`Gate.AmdocsCrm.Password`** properties.

   Note : The **`Gate.AmdocsCrm.UserName`** and **`Gate.AmdocsCrm.Password`** property values must be the same as the user name and password set for the Amdocs Web Application, TestCboInstallCpp.
5. Add a right click – tool to the OMNIbus event list GUI for logging tickets by adding the following line to the OMNIbus SQL file:

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

To ensure that the gateway configuration is successful pick an alert and mark it for ticket creation, either by setting the LogTicket column value to 1 using the LogTicket tool or using the following command:

- “`nco_sql update alerts.status set LogTicket=1` where `Serial=Serial no of alert`”.

  Note : The first ticket will take time to create as the gateway is logging into Amdocs.

If the gateway is correctly configured, a ticket will be created.

Setting up bidirectional gateways

This section describes how to configure the Amdocs server so that changes made to Amdocs cases are sent to Netcool/OMNIbus.

To configure bidirectional notifications you need to configure the Amdocs rule manager to send notification to the ObjectServer when Amdocs CRM updates the status of tickets. To configure the Amdocs server to send these notifications to the ObjectServer, use the following example:

`curl -data-ascii case_id=[`*`Object ID`*`],status='[`*`Status`*`]' http://`*`host`*`:`*`ServerPort`*

Where *Object ID* is the case type of updates you want the Amdocs server to forward.

Where *status* is event status you want the Amdocs server to forward.

Where *host* is the gateway host specified in the `httpTransport.properties` file.

Where *ServerPort* is the server port specified in the `httpTransport.properties` file.

Mapping syntax

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 Amdocs CRM to the Status, Journal, and Details tables within the ObjectServer. Mapping the configuration files for the Gateway for Amdocs CRM must use the following syntaxes.

Use the following syntax when configuring the `amdocscrm.rdrwtr.tblrep.def` file:

```
REPLICATE ALL FROM TABLE 'alerts.status'
        USING MAP 'StatusMap'
        FILTER WITH 'LogTicket=1'
        AFTER IDUC DO 'TTState=0';

REPLICATE ALL FROM TABLE 'alerts.journal'
        USING MAP 'JournalMap';
```

Use the following syntax when configuring the `amdocscrm.map` file:

```
CREATE  LOOKUP SeverityTable (
        {5 , 'High'})
DEFAULT = 'Medium' ;

CREATE  LOOKUP StatusTable (
        {0 , 'Solved Pending Confirmation'})
DEFAULT = '' ;


CREATE MAPPING StatusMap
(
 'alt_first_name' = 'Netcool' ON INSERT ONLY,
 'alt_last_name'  = 'OMNIBUS' ON INSERT ONLY,
 'phone_num'      = '(000)000-0000)' ON INSERT ONLY,
 'case_type_lvl1' = 'Incident' ON INSERT ONLY,
 'creation_time'  =  TO_TIME('@FirstOccurrence') ON INSERT ONLY,
 'case_reporter2contact:contact:first_name' = 'Anonymous' ON INSERT ONLY,
 'case2address:address:address'= '111 Main Street' ON INSERT ONLY,
 'case_reporter2site:site:name' = 'Unknown' ON INSERT ONLY,
 'respsvrty2gbst_elm' = Lookup('@Severity','SeverityTable'),
 'title' = "Netcool Alert:" + '@Summary' ON INSERT ONLY,
 'STATUS' = Lookup('@Severity','StatusTable'),
 'TTNumber'         =    '@TTNumber'
);

CREATE MAPPING JournalMap
(
        'Chrono'                =       '@Chrono',
        'description'           =       TO_STRING('@Text1') + TO_STRING('@Text2')
+ TO_STRING('@Text3'
);
```

The gateway handles mapping relationships in the `amdocscrm.map` file by populating the mapping field with the information gathered by the parameters set after the colon. For example

`case_reporter2contact:contact:first_name' = 'Anonymous' ON INSERT ONLY,`

In this example the `casereporter2contact` field is populated by the contact table row mapped to the filter `first_name='Anonymous'`, thereby populating the `casereporter2contact` field with anonymous.

Startup command file

The startup command file contains a set of commands that the gateway executes each time it starts.

You can specify the location of the startup command file using the generic Netcool/OMNIbus **`Gate.StartupCmdFile`** property.

The default startup command file is located in the following directory: `$OMNIHOME/gates/amdocscrm/amdocscrm.startup.cmd`

The default startup command file contains example commands. You should make a copy of the default file for future reference.

You can use the following commands within the startup command file:

- `SHOW PROPS` - Use this command to display the current configuration of the gateway by listing all properties and their values.
- `GET PROPERTY 'property_name'` - Use this command to return the value of the property specified in *property_name* from the gateway properties file.
- `SET PROPERTY 'property_name' TO ( 'string' | integer | TRUE | YES | FALSE | NO );` - Use this command to set the value of the property specified in *property_name* in the gateway properties file.
- `SET LOG LEVEL TO` - Use this command to set the level of message logging for the gateway. This command can take the following values: `fatal, error, warn, info` or `debug`. The default logging level is `warn`.

These commands can also be entered using the SQL interactive interface (`nco_sql`). For more information about using the SQL interactive interface, see the *IBM Tivoli Netcool/OMNIbus Administration Guide*.

For more information about the startup command file, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

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/amdocscrm/amdocscrm.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 4. Optional replication commands	
Command	**Description**
`FILTER WITH 'filter'`	Filters the database rows selected for replication, where *filter* 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 *order* would be `'FirstOccurrence'`.
`AFTER IDUC DO 'afteriduc'`	Updates replicated rows, where *afteriduc* 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'.

Gateway Properties

The following tables describe the properties available with the Amdocs Gateway. 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* (SC23-6387).

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

Gateway Reader-Writer properties

The following table lists the available gateway reader-writer properties.

Table 5. Gateway Reader-Writer properties		
Property name	**Command line option**	**Description**
`Gate.RdrWtr.Debug` *boolean*	-debug *boolean*	Use this property to enable the logging of gateway reader debug messages. The default is TRUE.

Table 5. Gateway Reader-Writer properties (continued)

Property name	Command line option	Description
Gate.RdrWtr.Description *string*	`-description` *string*	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 `Gateway Reader`.
Gate.RdrWtr.Details TableName *string*	`-detailstblname` *string*	Use this property to specify the name of the details table that the gateway reads. The default is `alerts.details`.
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 TRUE. **Note :** The gateway attempts to fail back with the frequency specified by the **Gate.RdrWtr. FailbackTimeout** property.
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. **Note :** The gateway attempts to fail back to the primary ObjectServer only if the **Gate.RdrWtr. FailbackEnabled** property is set to TRUE.

Table 5. Gateway Reader-Writer properties (continued)		
Property name	**Command line option**	**Description**
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`. **Note :** 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.JournalTableName *string*	`-journaltblname` *string*	Use this property to specify the name of the journal table that the gateway reads. The default is `alerts.journal`.
Gate.RdrWtr.LogOSSql *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.

Table 5. Gateway Reader-Writer properties (continued)

Property name	Command line option	Description
Gate.RdrWtr.Password *string*	`-password` *string*	Use this property to specify the password associated with the user specified by the **Gate.RdrWtr.Username** property. The default is " " **Note :** If the ObjectServer from which the gateway reads alerts is running on Tivoli Netcool/OMNIbus V7, 7.1, 7.2, or 7.2.1 this password must be encrypted by the `nco_g_crypt` utility. **Note :** If the ObjectServer from which the gateway reads alerts is running on Tivoli Netcool/OMNIbus V7.2.1 or greater in FIPS 140-2 mode, this password must be either plain text or encrypted using the `nco_aes_crypt` utility. For more information, see the *IBM Tivoli Netcool/OMNIbus Administration Guide*.
Gate.RdrWtr. ReconnectTimeout *integer*	`-reconntimeout` *integer*	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`.
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. StatusTableName *string*	`-statustable` *string*	Use this property to specify the name of the status table to which the gateway writes. The default is `alerts.status`.
Gate.RdrWtr. TblReplicateDefFile *string*	`-tblrepdeffile` *string*	Use this property to specify the path to the table replication definition file. The default is `$OMNIHOME/gates/amdocscrm/amdocscrm.rdrwtr.tblrep.def`.

Table 5. Gateway Reader-Writer properties (continued)		
Property name	**Command line option**	**Description**
Gate.RdrWtr.Username *string*	`-username` *string*	Use this property to specify the user name used to authenticate the ObjectServer connection. The default is `root`.

Mapping properties

The following table lists the available mapping properties.

Table 6. Mapping properties		
Property name	**Command line option**	**Description**
Gate.Mapper.Debug *boolean*	`-mapperdebug` *boolean*	Use this property to enable the logging of mapper debug messages. The default is TRUE.
Gate.Mapper.Forward HistoricDetails *boolean*	`-mapperforhistdtls` *boolean*	Use this property to specify whether the gateway forwards all historic details on converted update. The default is FALSE.
Gate.Mapper.Forward HistoricJournals *boolean*	`-mapperforhistjrnl` *boolean*	Use this property to specify whether the gateway forwards all historic journals on converted update. The default is FALSE.

Java properties

The following table lists the available Java™ properties.

Table 7. Java properties		
Property name	**Command line option**	**Description**
Gate.Java.Arguments *string*	`-javaargs` *string*	Use this property to specify the arguments to use when starting Java. The default is `-Xmx1024m`.
Gate.Java.ClassPath *string*	`-javaclasspath` *string*	Use this property to specify the environment variable used to store the location of the Java libraries. The default is `$CLASSPATH`.

Table 7. Java properties (continued)		
Property name	**Command line option**	**Description**
Gate.Java.Debug *boolean*	-javadebug *boolean*	Use this property to enable the logging of Java debug messages. The default is TRUE.
Gate.Java.LibraryPath *string*	-javalibrarypath *string*	Use this property to specify the location of the Java libraries that will be set in the environment variable specified by the **Gate.Java.ClassPath** property. The default is "".

Ticket Abstraction Layer (TAL) properties

The following table lists the available TAL properties.

Table 8. TAL properties		
Property name	**Command line option**	**Description**
Gate.TAL.NumberOfWriterThreads *integer*	-ttnumber_of_threads *integer*	Use this property to specify the number of writer threads writing to the target system. The default is 15.
Gate.TAL.FilterRepeatedValues *boolean*	-filter_repeated_threads *boolean*	Use this property to specify whether the gateway filters repeated value. The default is TRUE.
Gate.TAL.SafMode *integer*	-safmode *integer*	Use this property to specify whether the gateway uses the store and forward function when sending events. This property takes the following values: 0: The gateway does not run SAF mode. 1: The gateway uses SAF mode to record the last update change. 2: The gateway uses SAF mode to record updates based on IDUC granularity. The default is 0.

Table 8. TAL properties (continued)

Property name	Command line option	Description
Gate.TAL.SBScriptFileName *string*	`-sbscriptfilename` *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/amdocscrm/amdocscrm.script`. **Note :** If you want to stop the gateway polling requests, you must leave this property blank.
Gate.TAL.ResynchMode *integer*	`-resynchmode` *integer*	Use this property to specify whether the gateway resynchronizes on startup. This property takes the following values: 0: The gateway does not resynchronize on restart. 1: The gateway resynchronizes the status table on restart. 2: The gateway resynchronizes the status and journal tables on restart. The default is `0`.
Gate.TAL.Retry *boolean*	`-retry` *boolean*	Use this property to specify whether the gateway reattempts to poll Amdocs CRM after the occurrence of a recoverable error. The default is `FALSE`.
Gate.TAL.RetryAttempts *integer*	`-retryattempts` *integer*	Use this property to specify the number of times the gateway retries polling Amdocs CRM after the occurrence of a recoverable error. The default is 5.
Gate.TAL.TTJournalChronoFieldName *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.TTNumberFieldName *string*	`-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`.

Table 8. TAL properties (continued)		
Property name	**Command line option**	**Description**
Gate.TAL.TTState FieldName *string*	`-ttstatefieldname` *string*	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`.
Gate.TAL.Unrequired ColumnList *string*	`-unrequiredcolumnlist` *string*	Use this property to specify a list of column names that will not be used in order to determine if the update should be discarded in the case of a repeated value. **Note :** This is property is used with the hash code filtering properties specified in the **Gate.TAL.Filter RepeatedValues** property. The default is `""`.

Gateway specific properties

The following table lists the available Amdocs CRM properties.

Table 9. Gateway-specific properties		
Property name	**Command line option**	**Description**
Gate.AmdocsCrm. CloseOnDeleteList *string*	`-amdocscloseondeletelist` *string*	Use this property to specify whether a ticket in Amdocs CRM is closed if the 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'`. For example: `Gate.AmdocsCrm. CloseOnDeleteList : "STATUS='Solved Pending Confirmation'"` The default is `""`.

Table 9. Gateway-specific properties (continued)

Property name	Command line option	Description

Gate.AmdocsCrm.DispatchQueueFieldName *string*	-amdocscrmdispatchqueuefieldname *string*	Use this property to specify the Amdocs CRM queue to which the gateway sends events.
		The default is "".
		Note : The value specified in this property must have a corresponding entry in the amdocscrm.map file. Use the following example to add the correct entries:
		`CREATE LOOKUP QTable ( {5,'CriticalQueue'} ) DEFAULT 'DefaultQueue';`
		`'QueueName' = Lookup('@Severity','QTable) ON INSERT ONLY,`
		Then set the **Gate.AmdocsCrm.DispatchQueueFieldName** property to this corresponding value:
		`Gate.AmdocsCrm.DispatchQueueFieldName : 'QueueName'`

Gate.AmdocsCrm.ErrCodeFieldName *string*	-amdocserrcodefieldname *string*	Use this property to specify the ObjectServer field to which the gateway sends the error code that identifies the nature of the Amdocs CRM error. The default is AmdocsErrCode.
Gate.AmdocsCrm.HttpServerPropsFile *string*	-amdocshttpserverpropsfile *string*	Use this property to specify the location of the properties file for the HTTP server to which the Amdocs server sends case updates. The default is $OMNIHOME/gates/amdocscrm/httpTransport.properties.
Gate.AmdocsCrm.TransientErrorsList *string*	-amdocscrmtransienterrorslist *string*	Use this property to specify a list of transient Amdocs errors. The default is '(?s).*145752277.*,(?s).*145752384.*'
Gate.AmdocsCrm.RetryInterval *integer*	-amdocsretryinterval *integer*	Use this property to specify the interval (in seconds) between successive polls of the Amdocs CRM. The default is 30.

Table 9. Gateway-specific properties (continued)		
Property name	**Command line option**	**Description**
Gate.AmdocsCrm.Password *string*	-amdocsuserpassword *string*	Use this property to specify the Amdocs CRM password associated with the user name specified by the **Gate.AmdocsCrm.Username** property. The default is sa.

Gate.AmdocsCrm.StatusFieldName *string*	-amdocsstatusfieldname *string*	Use this property to specify the name of the mapper field used to populate the STATUS workflow attribute in Amdocs CRM tickets.
		The default is STATUS.
		Note : If the value of the mapper field is empty, the alert is not forwarded to the Amdocs CRM 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 amdocscrm.map file:
		`CREATE LOOKUP StatusTable (`
		`  {0, 'RESOLVED'})`
		`DEFAULT= '';`

Gate.AmdocsCrm.UserName *string*	-amdocsusername *string*	Use this property to specify the name of the user authorized to log on to Amdocs CRM. The default is sa.
Gate.MapFile *string*	-mapfile *string*	Use this property to specify the location of the map file used by the gateway. The default is $OMNIHOME/gates/amdocscrm/amdocscrm.map.
Gate.StartupCmdFile *string*	-startcmdfile *string*	Use this property to specify the location of the start up command file. The default is $OMNIHOME/gates/amdocscrm/amdocscrm.startup.cmd.

Running the gateway

To run the Gateway for Amdocs CRM, type the following command:

```
$OMNIHOME/bin/nco_g_amdocscrm
```

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
```

Error messages

This section describes the error messages that may be issued by the Gateway for Amdocs CRM.

Table 10. Error messages

Error	Description	Action

`Could not start Http server: %s`
`Could not create Http server: %s`
`Error stopping HTTP server(s)`
	An error occurred while the gateway was trying to use the HTTP server.	Check that the gateway can connect to the HTTP server.
		Check the **Gate.AmdocsCrm.HttpServerUserName**, **Gate.AmdocsCrm.HttpServerPassword** and**Gate.AmdocsCrm.HttpServerPropsFile** properties are correctly specified.

`Gateway startup failure due to cbo session creation failure: %d : %s`	An error occurred while the gateway was starting due to the Amdocs CPO API interface.	Check the Amdocs CPO API is configured correctly.
`Gateway startup failure: %s`	An error occurred while the gateway was starting.	Check that the gateway configuration file is correctly configured.
`Invalid close on delete property setting Name:%s`	An error occurred while attempting to feedback close details to the ObjectServer.	Check the **Gate.AmdocsCrm.CloseOnDeleteList** writer attribute is correctly specified.

Table 10. Error messages (continued)

Error	Description	Action
`Unable to parse the Close on delete list:%s`	An error occurred while attempting to feedback close details to the ObjectServer.	Examine error messages previous to this message to determine problem cause. Check that the gateway can connect to the HTTP server. Check the **`Gate.AmdocsCrm.CloseOnDeleteList`** writer attribute is correctly specified.
`Unable to update ticket for alert %{alert details} reason: %{cbo reason}`	An error occurred while updating a ticket due to the Amdocs CPO API.	Check that the Amdocs CBO API is correctly configured.
`Unable to process the error handling: %s`	An error occurred while updat	Check that the gateway mapping file is correctly configured.
`Unable to create ticket for alert %{alert details} reason: %{cbo reason}`	An error occurred while creating a ticket due to the Amdocs CPO API.	Check that the Amdocs CBO API is correctly configured.
`Unable to start the amdocs notification processor: %s`	An error occurred while starting the Amdocs notification process	Check that the gateway configuration file is correctly configured.

Appendix A. Notices and Trademarks

This appendix contains the following sections:

- Notices
- Trademarks

Notices

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

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

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

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

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

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

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

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

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

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

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

IBM Corporation
Software Interoperability Coordinator, Department 49XA

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

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

Trademarks

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

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

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

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

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

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

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

Part Number:

(1P) P/N:

SC22-5410-02