Nimsoft® Unified Monitoring™ Portal

Service Desk Integration Guide

2.1.1

Legal Notices

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Nimsoft Corporation disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Nimsoft Corporation shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Nimsoft Corporation and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Nimsoft Corporation as governed by United States and international copyright laws.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Nimsoft Corporation's standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Trademarks

Adobe®, Acrobat®, Acrobat Reader®, and Acrobat Exchange® are registered trademarks of Adobe Systems Incorporated.

Intel® and Pentium® are U.S. registered trademarks of Intel Corporation.

Java(TM) is a U.S. trademark of Sun Microsystems, Inc.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Netscape(TM) is a U.S. trademark of Netscape Communications Corporation.

Oracle® is a U.S. registered trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of the Open Group.

Contact Nimsoft

For your convenience, Nimsoft provides a single site where you can access information about Nimsoft products.

At http://support.nimsoft.com/, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- Nimsoft Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about Nimsoft product documentation, you can send a message to support@nimsoft.com.

Contents

Appendix A: Using the CMDB Gateway Probe 29

Chapter 1: Introduction

This guide tells you how to configure the Nimsoft Service Desk and My Tickets where the Nimsoft Monitoring Solution (NMS) server is hosted on premise and the Service Desk server is provided as a SaaS application. Service Desk and My Tickets are accessed via the Unified Monitoring Portal (UMP).

This section contains the following topics:

Service Desk Install Prerequisites (see page 7)

Service Desk Install Prerequisites

The Service Desk and My Tickets portlets depend on the following environment to be in place and fully functional:

- NMS server 5.1 or later
- Nimsoft Infrastructure Manager 3.86 or later
- Nimsoft UMP 2.1.1 or later
- Nimsoft Service Desk 6.0.5 or later
- Outbound access to port 443 for probes installed on the NMS server

Note: Not all web browsers are supported. See the UMP Release Notes for details, and other important information.

Chapter 2: Configure the Portlets

This section tells you how to install and configure the Service Desk and My Tickets UMP portlets.

This section contains the following topics:

Modify the wasp Probe Configuration

The wasp probe configuration must be modified to contain the URL of the Service Desk instance and the slice_token for authentication on the Service Desk server.

To modify the wasp probe configuration

1. Open Infrastructure Manager.
2. Click on the robot for the primary hub in the tree view.
3. Click the wasp probe to select it in the top right pane.
4. Press CTRL and right-click, then choose Raw Configure from the popup menu.
5. Expand the webapps folder in the tree view of the Raw Configure window.
6. Click servicedesk in the right pane.
7. Click the url key to select it, then click the Edit Key button.
8. Enter the URL for the Service Desk instance in the Enter new value field.

   The URL should be of the form http://<host>[:<port>]/<instance name>.
9. Click OK.
10. Click the slice_token key to select it, then click the Edit Key button.
11. Enter the value for the slice_token setting in the Enter new value field.

    The slice_token is a key that is generated automatically when the customer's "slice" is provisioned in Service Desk. If you do not know the slice_token for the customer, ask the support team that set up the customer's instance of Service Desk.
12. Click OK, then close the Raw Configure window.

Import the ServiceDesk.lar File

The Service Desk is designed to be displayed on its own page, and cannot be added to a page with other portlets. Because of this, it is not available on the UMP Add Applications menu as other portlets are. To create a page with the Service Desk you must import the ServiceDesk.lar file.

To import the .lar file

1. Point your browser to http://<umpServer>/servicedesk/jsp/get_lar.jsp and download the ServiceDesk.lar file.
2. Log into UMP.
3. Click the arrow in the upper right (next to the user name) and choose Control Panel from the drop-down menu.
4. Choose My Pages, then click the Private Pages tab.
5. Click the Export/Import tab, then click the Import tab.
6. Browse to the ServiceDesk.lar file, check User Preferences, and click Import.

Create Users

To use Service Desk and My Tickets, you must create users with the same names in both NMS and Service Desk. The following sections tell you how to create users in NMS and in Service Desk.

Chapter 3: Install and Configure the Service Desk Gateway

This section tells you how to install and configure the Service Desk Gateway.

This section contains the following topics:

Overview of the Nimsoft Service Desk Gateway Probe

The Nimsoft Service Desk Gateway lets you automatically or manually create incidents from Nimsoft Monitors Alarms. It supports bi-directional synchronization of alarms and incidents.

When alarms are assigned to the Nimsoft user or are automatically assigned, the nsdgtw probe creates a new incident in Nimsoft Service Desk. When these incidents are closed in Nimsoft Service Desk, the nsdgtw probe automatically acknowledges, in the Infrastructure Manager, the alarms from which these incidents were created.

In Nimsoft Service Desk, all the alarm attributes are available in the incident's Symptom description and Symptom details fields. Double-click the Service Desk incident to view its details.

When the operator using Nimsoft Service Desk has managed the incident and changed the status to **closed**, the gateway probe acknowledges the alarm.

The gateway probe lets you test the network access to Nimsoft Service Desk, or test the login session on Nimsoft Service Desk with verification of the user name and password.

Installation and Initial Setup

This section covers the deployment of the Service Desk Gateway probe (nsdgtw) and initial setup of the Nimsoft Server for its use.

Probe Installation

To install (deploy) the Service Desk Gateway probe:

1. Check that you have java_jre package v1.6 and robot_update v4.10. If you are using Nimsoft Server 5.1 or later, the correct java_jre package is preinstalled with the server. You can download the latest versions of the robot_update, java_jre, and jre_solaris packages from the Nimsoft Archive at http://support.nimsoft.com/, as is the Service Desk Gateway probe (nsdgtw) itself.
2. Open the Infrastructure Manager and select Archive.
3. Select the nsdgtw probe either locally or from the internet Archive, and drop it into Infrastructure Manager Archive.
4. After the probe is dropped into the Infrastructure Manager Archive, drag it from there and drop it on the robot where you want to deploy the probe.
5. After the probe has finished being deployed to the robot, click the Gateway category in that robot. You should see the deployed probe with a green mark beside it, indicating that probe is running.

Next you need to create the Nimsoft user from the Infrastructure Manager. See the Create a Nimsoft User (see page 12) section for details.

Create a Nimsoft User

The probe requires a Nimsoft user for its operation.

To create the required Nimsoft user:

1. Launch Infrastructure Manager.
2. In Infrastructure Manager, click **Security > User Administration**.
3. Right click inside the User Administration screen and click **New User** menu item. The New User dialog appears.
4. In the New User dialog, specify the new user's login ID in the **User** field.
5. Click the **Set Password** button.
6. Enter the password, retype it and click the **OK** button.
7. All other fields are optional. Click **OK** to create the specified new user.

Make a note of this user ID and password for later reference.

Note: This Nimsoft user is used only for assigning alarms generated in Infrastructure Manager.

Configure Offline Management

Offline Management lets a user take care of the alarms assigned by the Service Desk Gateway probe while the Nimsoft Service Desk server is down.

The probe contacts the Nimsoft Service Desk server at regular intervals. If the Nimsoft Service Desk server is down, nsdgtw halts until the server is restored.

When the Nimsoft Service Desk server resumes operation, the nsdgtw probe restarts. At this point, if Offline Management is enabled, the probe checks for the alarms assigned to the Nimsoft user in NAS while the Nimsoft Service Desk server was down. The probe fetches the list of alarms assigned to the Nimsoft user from NAS and compares it with the list in the configuration file. The probe creates incidents for those alarm IDs that are found in the NAS list but not in the configuration file list. The incidents are in Service Desk, as assigned from nsdgtw probe.

To disable or enable the Offline Management mode

1. Hold the Shift key down and right-click the nsdgtw probe in the Infrastructure Manager.
2. On the resulting pop-up menu, click the **Raw Configure** menu item.

   Note: Take care when using the "Raw Configure" tool. It is similar to the MS-Windows registry editor and has no error checking. Be sure that any changes you make are valid before you continue.
3. Select **setup** folder.
4. Select the **disable_offline_managment** key.
5. Click the **Edit Key** button.
6. To *disable* Offline Management, set the key to a value of **1**.

   To *enable* Offline Management, set the key to a value of **0**.

   Note: By default, the key is set to **1**.

   Modify the key value as required and click the **OK** button.
7. After configuring the settings, restart the nsdgtw probe to effect the changes.

Probe Configuration

This section describes how to configure the Service Desk Gateway probe.

Basic Configuration

To configure the Service Desk Gateway probe, double-click the nsdgtw entry in the Infrastructure Manager. This launches the gateway configuration interface.

Note: You must click the **Apply** button to activate any changes you make in the probe configuration.

The configuration interface has the following fields:

Log Level

Set the minimum acceptable level of detail to be written to the log file. Log as little as possible during normal operation to minimize disk consumption. You may want to increase the amount of detail when debugging, but be sure to reset it to a minimal level when finished.

nas address

Specify the address of the Nimsoft Alarm Server in this format:

/<Domain>/<Hub>/<Robot>/nas

Note: You must specify the local Nimsoft Alarm Server. In addition, the address is case sensitive.

Server URL

The URL of the web service for the Nimsoft Service Desk server

Username

User defined in the Nimsoft Service Desk.

Password

Enter the password for the user in Nimsoft Service Desk.

Test (button)

When you click the **Test** button, the gateway attempts to log in on the Nimsoft Service Desk server, using the specified login credentials. If the attempt is successful, the indicator turns green; otherwise, the indicator turns red.

Assigned User

Specify the assigned user (Nimsoft User) created in NAS.

Requester Name

Specify the name of the requester which is created in the Nimsoft Service Desk **server –> Application Setup –> Manage Contacts –> "Created By"** field.

Check interval

Specify the interval between checks for closed incidents in the Nimsoft Service Desk. When the operator in the Nimsoft Service Desk has managed the incident and changed the status to closed, the Service Desk Gateway probe (nsdgtw) clears the alarm, based on the **nimid**.

Check Now (button)

Click this button to check for closed incident in Nimsoft Service Desk in real time.

Time Zone

Specify the time zone code (such as GMT, IST, PST, etc.) to be used to store the time value. The probe can be configured to use a time zone other than the server's time zone. Using this field ensures that the nsdgtw probe and the server are synchronized with respect to time zones.

Note: The default time zone is GMT.

Incident ID Custom Field

Enter the custom field (custom1…custom5) in which the Incident ID generated on the server is to be displayed.

Auto Assign Alarms

This option enables the **Alarm Filters** button.

Edit Alarm Severity (button)

Click this button to edit the Alarm Severity level as required. Clicking this button raises Alarm Severity dialog screen. Select the desired severity level for respective alarms from the drop down list. The selected severity level is mapped with the Severity field of Nimsoft Service Desk.

Note: By default, equivalent Alarm Severities are selected.

Fields Mapping (button)

This button raises the Fields Mapping dialog. You can use this to map Service Desk custom fields with Nimsoft alarm fields. To edit an existing mapping, double click the existing record, select the new Service Desk Custom Fields for mapping, and click the Update button. Your changes will be saved.

Note: The **Alarm Fields** selector is disabled while editing.

To delete an existing mapping, select the desired record and click the **Remove** button.

Alarm Filters (button)

Select the alarm filter criteria based on which Nimsoft Service Desk incident should be created automatically from Nimsoft Server alarms . This field is enabled only when Auto Assign Alarm is selected.

Assigning Values to Custom Fields

To assign values to custom fields in infrastructure manager:

1. Right click selected alarm.
2. Select **Select Custom Field** option from the pop-up.
3. After hovering on **Select Custom Field** five custom fields will appear.
4. Select anyone to whom the value is to be assigned.
5. The **Set Custom Field Value** dialog will appear. Enter a value and press **OK**.

Chapter 4: Configure Service Desk

NMS alarms automatically create incident tickets in the UMP Service Desk portlet. Operators can also manually create an incident ticket from an alarm in the UMP Alarm Console. When an incident ticket is closed in Service Desk, the corresponding alarm in the NMS Alarm Console is also closed.

In order for this integration to work, you must configure a Web Services user account in Service Desk.

If you want, you can also change the alarm data fields that are imported into the incident ticket. To do this, you modify the custom fields in the Service Desk Incident Ticket Template.

This section contains the following topics:

Create a Web Services Account

To use Web Services to access Nimsoft Service Desk, a Web Services user account must be configured in Service Desk. The Web Services client account must be configured as a contact record in Service Desk.

Note: To configure and use the Web Services user account, you must have the required Web Services license.

To configure a Web Services account

1. Click the Manage Contacts link in the Application Setup Section of the Service Desk Navigation Panel.

   The Manage Contact form is displayed.

2. Enter information in the First Name and Last Name fields.

   This will be the Display Name for the client. Both fields are required.

3. Enter additional information if desired, or leave the rest of the contact record form blank.

4. Click Apply Changes.

   The new contact record is created.

   The Contact Details, Location, Open Items, and other tabs are now displayed on the form. The Enable Login check box, Out of Office check box, and other related fields are enabled.

5. Click the Enable Login check box in the contact record section.

   This allows the Web Services User access to the application.

6. Relate the user to an organization in the Location tab of the contact record.

   This will be the primary organization of the user. You can relate multiple organizations, but only one organization can be marked as primary.

7. Do the following in the Application tab of the contact record:

   - Assign System User ID for the Web Services User
   - Assign License type as Web Services
   - Check the Disable Service Feedback check box as this contact is not needed to participate in the Service Feedback process.

8. Click Apply Changes.

   Your changes to the Web Services user account are saved.

Modify Custom Fields in the Incident Ticket Template (Optional)

Nimsoft Service Desk is preconfigured to add alarm data as custom fields in the Incident Ticket Template. Adding alarm data to the Incident Ticket Template means the alarm data is available when viewing incident tickets.

If you want different alarm data displayed in incident tickets, you can follow the steps in the following sections to modify the custom fields in the Incident Ticket Template. These sections tell you how to create attributes with names that match alarm data fields, then add those attributes as custom fields to the Incident Ticket Template.

Note: Attributes cannot be deleted once they are configured and used in templates.

The alarm data listed in the following table is added to the Incident Template Ticket by default.

Attribute Name	Attribute Type
Affected Device	Text
Alarm Received	Date Time
Alarm ID	Text
Alarm Count	Number
Suppression Key	Text
Probe Name	Text
Device Type	Text
Robot	Text
Hub	Text
Domain	Text
Time Origin	Date Time
Time Arrival	Date Time
Time Assigned	Date Time
Notes	Text

Create Attributes for Alarm Data Fields

To create attributes for alarm data

1. Click the Manage Attributes link in the Application Setup section of the Service Desk Navigation Panel.

   The Manage Attributes form is displayed. A list of existing attributes is displayed in the table.

2. Click Create New.
3. Enter the name of the attribute you want to add in the Attribute Name field.

   The name of the attribute must match the name of an alarm data field.

4. Set the Sort Order to an integer.

   This controls the order in which attributes are displayed in templates.

5. Choose the appropriate attribute type from the Attribute Type drop-down list.
6. Click Apply Changes.

   The attribute record is saved.

7. Add attributes for any other alarm data fields you want.

Add the Attributes to the Incident Ticket Template

To add the attributes to the Incident Ticket Template

1. Click the Manage Custom Fields link in the Application Setup section of the Service Desk Navigation Panel.

   The Manage Custom Field Templates form is displayed. The Custom Field Templates are listed in the table.

2. Click Form Name in the filter bar and choose Incident.
3. Click Template Type in the filter bar and choose System Defined.
4. Click Refresh.

   The default Custom Fields Template for Incident Tickets is listed in the table.

5. Click the row for Incident in the table.

   The Incident Ticket Template is displayed in the bottom pane.

6. Click the lookup button next to the Attribute Name field.

   The Attribute Name List is displayed in a dialog.

7. Click the name of an attribute you added in the previous section.

8. Click Save Attributes.

   The attribute is added to the Incident Ticket Template.

9. Add all of the attributes you created for alarm data fields.

10. Click Apply Changes.

    The alarm data fields will now be captured when a ticket is created manually from an alarm or created automatically by an alarm generated by a monitoring device.

Chapter 5: Install Unified Reporter

Install the Unified Reporter version 1.6.1 or later. This provides you with a number of pre-configured UMP reports, including reports with information about Service Desk incident tickets related to the top 10 sources (devices, servers, applications, accounts, and service quality) generating NMS alarms.

The Unified Reporter software is available from the Downloads page at support.nimsoft.com. For installation instructions, see the *Unified Reporter Installation Guide*, also available from the Downloads page.

Note: In order for the pre-configured Service Desk reports to work you must have done the configuration steps for the portlets and Service Desk in the previous sections.

Chapter 6: Install and Configure the CMDB Gateway and Service Desk Adapter

This section describes how to install and configure the components required to synchronize Nimsoft CMDB data with Service Desk.

This section contains the following topics:

Install the CMDB Gateway Probe

To install the CMDB Gateway probe

1. Download the CMDB Gateway probe from the Nimsoft internet archive (http://support.nimsoft.com).
2. Optionally, download the *CMDB Gateway Probe Installation and Configuration Guide*, which is available with the probe. That document is reproduced here in the Using the CMDB Gateway Probe (see page 29) section.
3. Install the CMDB Gateway probe as instructed in the Using the CMDB Gateway Probe (see page 29) section.

If you plan to use the CMDB Gateway probe only for Service Desk synchronization, the specific steps for the necessary configuration appear in the Configure the CMDB Gateway to use the Service Desk Adapter (see page 27) section.

In addition, you can configure the CMDB Gateway probe for other purposes as desired; see the Using the CMDB Gateway Probe (see page 29) section for details.

Install the ServiceDesk Adapter for the CMDB Gateway

The Service Desk synchronization adapter consists of a command line application and an XSL transform.

To install the Service Desk adapter

1. Download the **sdsync.zip** file from the Nimsoft internet archive or Downloads section of the Nimsoft support site (http://support.nimsoft.com/).
2. Unzip this file into a subdirectory of the **cmdbgtw** directory called **sdsync**, as in this example:
   `probes/gateway/cmdbgtw/sdsync`

Next you must configure the CMDB Gateway probe to make use of the Service Desk synchronization adapter.

Configure the CMDB Gateway to use the Service Desk Adapter

The steps below configure the CMDB Gateway probe to make use of the Service Desk Adapter.

To configure the CMDB Gateway probe for the Service Desk adapter

1. Double-click the cmdbgtw probe in Infrastructure Manager to launch the probe's configuration interface.
2. Create an Export for the Service Desk Adapter, as follows:
   a. Right-click on the Export hierarchy, and choose New Export. The New Export dialog is displayed.
   b. Enter information in the fields as described below.

      Export Name

      A descriptive name. The name must be unique.

      Root Entity

      The Service Desk adapter requires you chose **CmComputerSystem** as the root entity.

      Database

      Choose a database connection.
3. Enter the following in the **XSLT Transform** field:
   `sdsync\ServiceDesk.xsl`
4. In the **Output Format** selection list, click on **XML**.
5. Use the **Schedule** check box to enable a scheduled export, and set **Run every** to **1 hour**.
6. Use the **Finished Program** field to specify the command line program, as follows:

   For Windows systems

   `sdsync\Import.bat -url <service desk url> -user <userid> -pass <password>`

   For Unix® systems

   `sdsync\Import.sh -url <service desk url> -user <userid> -pass <password>`

   The user you specify in this field must have a web service license in Service Desk; see Create a Web Services Account (see page 18) for details.

   Note: If you want to encrypt the password, use the **Add Property** button to add the **-pass** parameter to the **Finished Program** specification. For details, see the Using the CMDB Gateway Probe (see page 36) section
7. Use the **Include Type** check box to include everything in the Mapping hierarchy.
8. Set the **Change Marker** check box on the **changeTime** column of the **CmComputerSystem** property.

Troubleshooting the Service Desk Adapter

If you have concerns about the proper configuration of the Service Desk Adapter for the CMDB Gateway probe, check the following:

- Examine the **cmdbgtw** log, located in **probes/gateway/cmdbgtw**. If the log shows zero (0) entities exported, the the Service Desk adapter is receiving no data. Use the CMDB Gateway probe configuration interface to make the changes required to export data.
- After the first export, the **probes/gateway/cmdbgtw/sdsync** directory contains a file named **cmdbImport.log**.It contains the final output of the last run, which may provide clues to potential configuration errors.
- Examine the **probes/gateway/cmdbgtwCIBulkImportUtility0.log** log file. It contains full details of the last run, and any errors should be detectable in it.

Note: If the command line program fails for any reason, the CMDB Gateway probe will not change the Change Marker. The next export will still detect any changes that occurred after the last successful run.

Appendix A: Using the CMDB Gateway Probe

This section describes how to install and configure the CMDB Gateway probe. Note that this probe is not restricted for use with the Service Desk Adapter, and this section does not call out the specific configuration required for that purpose.

For explicit CMDB probe configuration instructions related to the Service Desk Adapter, see the Install and Configure the CMDB Gateway and Service Desk Adapter (see page 25) section.

This section contains the following topics:

Prerequisites and Supported Platforms

The CMDB Gateway probe and its configuration interface require the following:

- Nimsoft Monitor 5.1 or later
- Java 1.6.0.14 or later

Probe Installation

The CMDB Gateway probe installs via drag-and-drop from the Nimsoft Infrastructure Manager, and requires only the standard Nimsoft Java runtime package that is part of the NM server.

Using an Alternate JDBC Driver

If you want to use a JDBC driver other than the one provided with the CMDB Gateway probe, follow these steps:

1. Use the Infrastructure Manager to stop the probe.
2. Right-click on the probe in Infrastructure Manager, and choose **Edit**. from the menu
3. Modify the Arguments field by adding JDBC driver jar(s) to the **-cp** argument. For example, on Windows the result would be:

   ```
   -cp cmdbgtw.jar;lib/*.jar;c:\mystuff\mydriver.jar
   ```

   On UNIX®-based operating systems, be sure to use a colon character (**:**) as the path separator character.
4. Restart the probe.
5. In the probe configuration interface, modify or create a database connection so that the **Driver Class > Use** property is the fully qualified class name of your driver class.

Installing the Probe Configuration Interface

The CMDB Gateway probe configuration interface is automatically downloaded and installed by the Nimsoft Infrastructure Manager when you first attempt to configure the probe.

Note: The configuration interface may fail to start if more than one java_jre package is installed and any of the java_jre packages predates version 1.6.0_14. If this is the case, you must remove obsolete java_jre packages before you can successfully launch the configuration interface.

Probe Configuration

This section describes the configuration concepts and procedures for setting up and using the CMDB Gateway (cmdbgtw) probe.

Overview of Probe Configuration and Operation

There are two primary steps in configuring the CMDB Gateway probe.

1. First, you define one or more Database Connection objects for the databases you wish to export data from.
2. Then you create one or more Export objects. Each Export defines the type of data to export, the database to export from, data filters and transformations, and the output format.

Later sections explain the configuration process in detail.

A context-sensitive popup menu is available by right-clicking on a node in the Export hierarchy. It has self-descriptive choices.

Configuring Logging

The CMDB Gateway probe can be configured to log messages. To configure logging, click the **Options** button on the top toolbar to display the Probe Options dialog. It contains the following options.

Log Level

Determines the level of detail included in log messages. When you first use the probe or are actively troubleshooting a problem, set the log level to **Debug**. Once the probe is functioning properly, change the logging to **Normal Operations** to save disk space. Use the **Errors/Warnings Only** level only for rare situations when disk space is at a premium.

Max Log Size(MB)

Maximum size, in megabytes, of a single log file.

Max Logfiles

When a log file reaches the **Max Log Size**, the probe saves the old log into a file, empties the primary log file, and continues logging. This setting determines the number of saved log files to retain.

Creating and Configuring Database Connection Objects

To configure the CMDB Gateway probe, you first create one or more database connection objects using the configuration interface.

To Create a Database Connection Object

1. You can either edit the provided My Database connection or create a new connection. To create a new connection object, right-click on any node in the hierarchy, and choose New Database Connection. The new connection object appears in the hierarchy.
2. Click the node for the database connection you want to create or modify to select it. The configuration information for the database connection object appears in the right pane.
3. Enter the information in the fields as described below.

   Name

   Enter a name for the connection.

   User

   Database user name.

   Password

   Database password.

   Driver Class

   The Java JDBC driver class to use. Select a class from the Supported drop-down list.

   Note: JDBC drivers for Oracle, MySQL, and Microsoft SQL Server are included with the probe. You can run the probe using a driver that is not shipped with the probe. See Advanced Data Mapping (see page 41) for advanced configuration.

   SQL Dialect

   Choose a dialect that matches the vendor and version of your database. Once you choose a dialect, the URL Format field is populated with a sample JDBC connection URL.

   Connection URL

   Your JDBC connection URL.
4. Verify that the probe is running, and then click the **Test Connection** button. You are notified whether your connection information is valid.
5. The **Use For All Exports** button associates the current database connection with all defined Exports. This is useful if your database moves.

Creating and Configuring Exports

An Export is a group of settings for exporting data from a database. It includes the type of data to export, the database to export from, the output directory, and the format for the exported data.

To Create an Export

1. Right-click on the Export hierarchy, and choose New Export. The New Export dialog is displayed.
2. Enter information in the fields as described below.

   Export Name

   A descriptive name. The name must be unique.

   Root Entity

   Data type to export.

   Note: The CMDB Gateway probe supports exports from some of the Nimsoft SLM database schema. You can add support for missing entity types. See Advanced Data Mapping (see page 41) for advanced configuration options.

   Database

   Choose a database connection.

Configuring Export Options

Select the new Export in the hierarchy to see its full configuration in the right pane.

There are two major aspects of the Export configuration:

- The Setup section
- The Mapping section

Configuration Options in the Setup Section

The configuration options in the Setup section are as follows:

Name

A descriptive name.

Root Entity

Data type to export. You cannot change the root entity of an existing Export. You must create a new Export if you wish to use a new root entity.

Output Directory

The path, absolute or relative to the probe's installation directory, where the probe will write exported data.

Output File

A filename to use when writing Export data, for example **export.xml** or **scheduledEveryMinute.xml**. If you specify a filename, the probe writes to that file, overwriting its contents each time the Export runs. If you do not specify a filename, the probe generates a unique filename for you, using the Export name and the start time of the export. Each export generates a new file.

Database

Select the database connection to use.

XSLT Transform

Enter the path, absolute or relative to the probe's installation directory, that specifies an XSLT transform to run on the exported data. This is an optional feature that applies only when the output format is set to XML. The transform is applied after all root entries have been extracted from the database and therefore the XML includes the root entity parent element **<items>**.

Output Format

Select XML or CSV (Comma-Separated Values). Note that with CSV, only root entities are included in the output. This is because the CSV format can not represent structured relationships.

Schedule

This property lets you to run an Export on a repeating periodic schedule. See Scheduling Exports and Receiving Data (see page 40) for details.

Finished Program

Identifies an external script or program to run each time an export completes. You may provide an absolute path or a relative path (to the probe's installation directory). See section Scheduling Exports and Receiving Data (see page 40) for program argument details.

Custom Properties

Exports can be further customized by adding custom user-defined properties.

To Define a Custom Property

1. Click the **Add Property...** button in the Setup section
2. Provide values for the following fields:

   Property Name

   A unique name for your property.

   Description

   Textual description of your property

   Secure?

   If checked, the property becomes a secure string. A secure property is masked with asterisk characters ('*') in edit fields, and the property value is encrypted in the probe's configuration file.

When you click "OK", the new property appears at the bottom of the property list. To delete a custom property, click the red 'X' icon to the right of the property.

Custom properties are passed to the **Finished Program** as command line arguments. The properties are passed using the following syntax:

-<property_name_without_whitespace> <property value>

For example, if you define a property named **My CMDB Password** with a value of **pw12345**, the property and value are passed as follows tot he Finished Program:

```
-MyCMDBPassword pw12345
```

Using a "Finished Program" to Integrate with External Systems

A scheduled Export has an optional **Finished Program**, which can be any external script or program that is run each time an Export completes. You can provide an absolute path or a relative path (to the probe's installation directory) to the Finished Program.

The program will be passed the command line arguments listed below, all in this form:
`–[argname] [argvalue]`

-exportName

The configured name of the Export.

-startTime

The date and time the export started, in W3C date/time format (See http://www.w3.org/TR/NOTE-datetime, not affiliated with Nimsoft). The format is as follows:
`yyyy-MM-dd'T'HHmmss.SS'Z'`

-result

The text string '**ok**' if the export succeeded, or a string describing an error.

-nEntities

The number of root entities exported.

-outputFile

The complete path to the file containing the exported entities.

You can specify additional command line arguments when entering the Finshed Program property. These additional arguments are passed *before* the standard arguments outline above.

The following example Windows batch file uses the popular **cURL** program to upload the Export data to a server using HTTP PUT:

```
:parseargs
IF NOT "%1"=="" (
    IF "%1"=="-exportName" (
        SET exportName=%2
        SHIFT
        GOTO endif
    )
    IF "%1"=="-startTime" (
      SET startTime=%2
        SHIFT
        GOTO endif
    )
    IF "%1"=="-result" (
      SET result=%2
        SHIFT
        GOTO endif
```

```
    )
    IF "%1"=="-nEntities" (
      SET nEntities=%2
        SHIFT
        GOTO endif
    )
    IF "%1"=="-outputFile" (
      SET outputFile=%2
        SHIFT
        GOTO endif
    )
:endif
SHIFT
GOTO parseargs
)

if "%result%"=="ok" (
        curl -v --upload-file "%outputFile%" http://localhost/onComputers
)
```

There are several points to keep in mind about using a **Finished Program**:

- The **stdout** and **stderr** of your finished program is appended to the probes log file if the log level is **Debug** or higher.
- The **Test** button in the configuration user interface will also display the **stdout** and **stderr** of your finished program in the scrolling results window.
- On Windows operating systems you may specify a **.bat** file as your finished program without explicitly invoking **cmd.exe**
- To run a shell or Perl script on UNIX and LINUX operating systems you will need to explicitly call the interpreter. For example: **/bin/sh myTrigger.sh**
- If your Export is a scheduled export, the Export's change marker is updated only if the **Finished Program** returns **0** for success.

Configuring the Export Data Mapping

The Mapping section lets you fine tune the data to export.It is subdivided into two parts. The entity hierarchy is on the left. Select an entity in the hierarchy to see its details and options on the right, as described below:

Include Type

If you are not interested in related entities, deselect this check box. This turns off the inclusion of related entity types in the Export. You must include the root entity by definition. Excluded entity types are identified in the hierarchy by a red dot icon and grayed-out text.

XML Parent

The read-only XML element of the parent node of this entity type. For example, for the **CmDevice** type the parent is **devices**, resulting in this XML:

```
<devices>
  <device>...</device>
  <device>...</device
  ...
</devices>
```

The root entity type always has a parent XML element named **<items>**.

XML Name

The read-only XML element name for entities of this type.

Filter

A where clause query expressed in Hibernate Query Language. See Advanced Data Mapping (see page 41) for advanced configuration options.

Properties

This table shows the properties (also known as database columns) that comprise the entity type. The properties are as follows:

Name

Property name.

Column

Database column name.

XML Name

Name of the XML element that holds this property value. If the name starts with the **@** character, the property is written as an XML attribute of the entity's XML name.

For example, **CmComputerSystem** has property named **csId** with XML Name of **@id**. It also has a property named **csType** with XML Name of **csType**. The resulting XML is as follows:

```
<computer id="[the id]">
 <csType>[the type]</csType>
```

```
 ...
</computer>
```

Formula

A valid Hibernate Query Language column formula expression. Formulas are used to transform data during export. To edit the formula, double-click the property row. See Advanced Data Mapping (see page 41) for advanced configuration options.

Change Marker

Lets you specify a column in the root entity as a change indicator. (new entity, modified entity). Currently only columns with the custom **com.nimsoft.db.MyTimestampType** type mapping are supported. See the Scheduling Exports and Receiving Data (see page 40) section for details.

Mapping Notes

All date and time columns in the provided **.hbm.xml** files map using a custom type called **com.nimsoft.db.MyTimestampType**. This causes all dates and times to be written in W3C date and time format as detailed in http://www.w3.org/TR/NOTE-datetime (not affiliated with Nimsoft).

The full format is as follows:
`yyyy-MM-dd'T'HHmmss.SS'Z'.`

See Advanced Data Mapping (see page 41) for information on implementing your own custom data types.

Testing and Running Exports

To test or run an Export, select an Export in the tree to see the Test and Run buttons in the right pane.

Test

When you click **Test**, the probe performs the export but only provides results for the first entity from the database. The test result is displayed in the bottom right pane of the configuration interface window.

Run

When you click **Run**, the probe performs the export and writes the complete results to a file in the Export's configured **Output Directory**. The file is named according to the following pattern:

[export name]_[export date].xml

The export date is expressed in W3C date and time format as detailed in http://www.w3.org/TR/NOTE-datetime (not affiliated with Nimsoft).

Scheduling Exports and Receiving Results

The CMDB Gateway probe (cmdbgtw) includes features that let you schedule an Export to run periodically, and notify external systems when the scheduled Export is finished. If you enable scheduling on an Export, the probe runs that Export periodically according to the schedule you specify.

If the root entity you are exporting has a **timestamp** column that is updated when new entities are inserted or entities are modified, you can use a change marker to ensure that successive runs of an Export output only new or changed entities. To designate a column as a change marker, select the root entity, double-click the row in the Properties table for the column and check the **Change Marker** check box.

There are a few things to keep in mind:

- If you change the column used as the change marker, the next run of the Export exports all entities. Subsequent runs export only new or changed entities.
- A scheduled export that does *not* have a designated change marker column *always* export all entities.
- Currently only columns with the custom **com.nimsoft.db.MyTimestampType** type mapping are supported as change marker columns.

Advanced Data Mapping

Overview of Advanced Data Mapping

The CMDB Gateway probe uses the Hibernate open source relational data mapping framework. This section assumes a familiarity with Hibernate. For more information about Hibernate, see http://www.hibernate.org/ (not affiliated with Nimsoft).

The probe installation directory contains a subdirectory named **mappings/base**. The **base** subdirectory contains Hibernate mapping files for all entity types that the CMDB Gateway probe recognizes. (This version of the CMDB Gateway probe includes a small subset of the entire Nimsoft SLM database schema.)

An Export consists of the base mapping plus any customizations you configure. Customizations you can configure using the configuration interface include the following:

- Excluding related entity types
- Filtering collections of entities
- Specifying transform formulas for individual columns

During an export, the probe uses the base **.hbm.xml** files and the customizations to generate Export-specific **.hbm.xml** files in the **mappings/<mapping name>** directory. Note that Export-specific **.hbm.xml** files are generated only when changes are made to customizations.

Specifying Entity Filters

To create an entity filter, select an entity in the mapping hierarchy and enter a **where** clause expressed in Hibernate query language in the filter box. You must use database column names in your expression.

For example, if your root entity type is **CmComputerSystems** but you only want to export computer systems on a specific subnet, enter a filter such as this:

```
ip like '10.0.1.%'
```

The Hibernate reference documentation contains complete details on the Hibernate query language.

Changing the Base Mapping

To customize your Exports, change the base mapping by either modifying the **.hbm.xml** files in **mappings/base**, or by adding new **.hbm.xml** files to that directory. You must restart the probe and the configuration interface after making changes to the base mapping.

Here is a custom mapping example.

Custom Table Example

You have a custom table added to your Nimsoft SLM database schema called **COMPUTER_ANOMALY** that has a foreign key relationship with the **CM_COMPUTER_SYSTEM** table.

To export this custom data, create a Hibernate **.hbm.xml** file for your custom table, and modify the **CmComputerSystem.hbm.xml** file to have a relationship to this table, as follows.

MyComputerExtra.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated Aug 26, 2010 10:58:25 AM by Hibernate Tools 3.3.0.GA -->
<hibernate-mapping>
  <!-- note that we use entity-name and not class-name -->
  <class entity-name="ComputerAnomaly" node="anomaly" table="ComputerAnomaly">
    <id name="id" type="int">
      <column name="id" />
    </id>
    <property name="date" type="com.nimsoft.db.MyTimestampType">
      <column name="date" length="23" />
    </property>
    <property name="description" type="string">
      <column name="description" length="128" />
    </property>
  </class>
</hibernate-mapping>
```

Modifications to CmComputerSystem.hbm.xml

Add the relationship as follows:

```
<set name="anomalies" node="anomalies" table="ComputerAnomalies" inverse="true"
lazy="true" fetch="select">
    <key>
            <column name="computer_id" />
    </key>
   <one-to-many class="ComputerAnomaly" />
</set>
```

Your exported XML will then have the following form:

```
<computers>
  <computer>
    ...
    <anomalies>
      <anomaly id="theid">
        <date>thedate</date>
        <description>thedescription</description>
      </anomaly>
    <anomalies>
```

```
  </computer>
</computers>
```

Implementing Custom Property Types

Hibernate properties have a type that determines how Hibernate maps the database data to XML data. You can implement custom types to affect how data is output.

The CMDB Gateway probe uses this feature to output dates in W3C date and time format. For example, the **create_time** column in **CmComputerSystem** is mapped as follows:

```
<property name="createTime" type="com.nimsoft.db.MyTimestampType">
  <column name="create_time" length="23" not-null="true" />
</property>
```

MyTimestampType is a custom class that implements **org.hibernate.usertype.UserType**. It tells Hibernate to map date/time columns to **com.nimsoft.db.MyTimestamp** objects instead of to **java.sql.Timestamp objects**. **MyTimestamp** then overrides **toString()** to return a properly formatted date and time. Example code is available upon request.

There are a couple of things to keep in mind when using custom mapping:

- Use the **entity-name** property on a **<class>**, not the **class-name** property.
- You can map a table more than once as long as you use a unique **entity-name**. For example, you can have an **entity-name=Computer** mapping that includes all properties, and an **entity-name=ComputerChanges** mapping that includes only the **id** and **change_time**.

Nimsoft Callback Interface

The CMDB Gateway probe includes a custom callback called **runExport**. Other Nimbus participants can send this message to the CMDB Gateway probe to cause it to run the named Export.

runExport--Invoke the CMDB Export from Other Probes

This callback allows other Nimbus participants to invoke an Export by the CMDB probe.

runExport Parameters

exportName

Text name of the Export you assigned when creating the Export in the interface.

entityCountLimit

Limits the number of entities returned. Enter 0 for unlimited entries, or a positive integer for a specific number of entities. Defaults to 0.

outputFile

A full path to a file where export results should be written. If not specified, the probe will generate a unique file in the Export's Output Directory.

filterXml

See About filterXML (see page 45).

About filterXML

A filter defines selection criteria (that is, a WHERE clause) for an Export's root entity. If a filter is specified in runExport, it completely overrides any filter that might be configured via the configuration interface.

Filters are specified using the following XML format:

```
<filter>
            <name>MyFilter</name>
            <isEnabled>true</name>
            <expression>
                 change_time &gt; :pChangeTime OR cs_id > :pMinId
            </expression>
            <parameters>
                <parameter>
                    <name>pChangeTime</name>
                    <type>DATE</type>
                    <value>1970-01-01T00:00:00.000Z</value>
                </parameter>
                <parameter>
                    <name>pMinId</name>
                    <type>INT</type>
                    <value>44</value>
                </parameter>
             </parameters>
        </filter>
```

runExport Example Filter

This example filter selects computers whose **change_time** column has a date more recent than **1970-01-01T00:00:00.000Z**, or whose **cs_id** is greater than 44.

The one and only filter object is as follows:

name

The filter name. The parameter is required, but not really used in any important way.

isEnabled

Set to **true** or **false**

expression

The Boolean filter expression. You should use actual column names and not Hibernate property names. This property supports named parameters using HQL syntax (**:parametername**). Each named parameter requires a **<parameter>** entry in the **<parameters>** list.

parameters

Optional, set of named parameters

parameter

name

parameter name

type

One of:

- **STRING**
- **INT**
- **LONG**
- **BOOLEAN**
- **DOUBLE**
- **DATE**

value

Must be convertible to the parameter type. If type is **DATE**, the value must be in W3C date/time format (http://www.w3.org/TR/NOTE-datetime, not affiliated with Nimsoft). The format is as follows:

`yyyy-MM-dd'T'HHmmss.SS'Z'`

The following name/value pairs result from the above:

msg

Message indicating success or failure of export. Contains the string '**ok**' if successful, or contains an error message if it failed.

outfile

Path to the exported data file.

nEntitiesGenerated

Number of root entities generated.