Netcool Configuration Manager 6.4.2

Administration Guide

2023-4219-01

Note

Before using this information and the product it supports, read the information in “Notices” on page 151.

This edition applies to version 6.4.2 of IBM Tivoli Netcool Configuration Manager (5725-F56) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this publication

IBM Tivoli Netcool Configuration Manager provides network management and configuration capabilities. Netcool Configuration Manager - Base provides the configuration management capabilities for network devices, and Netcool Configuration Manager - Compliance provides extensive configuration policy thresholding capabilities.

The *IBM Tivoli Netcool Configuration Manager Administration Guide* guide describes administration tasks for IBM Tivoli Netcool Configuration Manager, such as how to set up user accounts, create and manage the OS registry, administer database and policy exports and imports, and perform housekeeping and security tasks.

Intended audience

This publication is for administrators who are responsible for the maintenance and availability of IBM Tivoli Netcool Configuration Manager.

What this publication contains

This publication contains the following sections:

Publications

This section lists publications in the Netcool Configuration Manager PDF document set. The prerequisite publications in the IBM Tivoli Network Manager IP Edition and IBM Tivoli Netcool/OMNIbus library are also listed here. The section also describes how to access Tivoli publications online and how to order Tivoli publications.

Netcool Configuration Manager PDF document set

The following documents are available in the Netcool Configuration Manager library:

- *IBM Tivoli Netcool Configuration Manager Installation and Configuration Guide*

  Describes how to install IBM Tivoli Netcool Configuration Manager. It also describes necessary and optional post-installation configuration tasks. This publication is for administrators who need to install and set up IBM Tivoli Netcool Configuration Manager.

- *IBM Tivoli Netcool Configuration Manager User Guide*

  Describes user tasks for IBM Tivoli Netcool Configuration Manager, such as how to access reports, use devices, and execute the different utilities to maintain and support Auto-Discovery. This publication is for users working with IBM Tivoli Netcool Configuration Manager.

- *IBM Tivoli Netcool Configuration Manager Administration Guide*

Describes administration tasks for IBM Tivoli Netcool Configuration Manager, such as how to set up user accounts, create and manage the OS registry, administer database and policy exports and imports, and perform housekeeping and security tasks. This publication is for administrators who are responsible for the maintenance and availability of IBM Tivoli Netcool Configuration Manager.

- *IBM Tivoli Netcool Configuration Manager Reference Guide*

  Contains reference information about IBM Tivoli Netcool Configuration Manager.

- *IBM Tivoli Netcool Configuration Manager API Guide*

  Provides information about how to use the Java API to programmatically access IBM Tivoli Netcool Configuration Manager.

- IBM Tivoli Netcool Configuration Manager NSM REST API Guide

  Describes the Service Management Interface API.

- *IBM Tivoli Netcool Configuration Manager Integration Guide*

  Describes how to integrate Netcool Configuration Manager with Tivoli Netcool/OMNIbus and Network Manager.

- *IBM Tivoli Netcool Configuration Manager Quick Start Guide*

  Gets you started with a typical installation for IBM Tivoli Netcool Configuration Manager.

- *IBM Tivoli Netcool Configuration Manager Release Notes*

  Gives important and late-breaking information about IBM Tivoli Netcool Configuration Manager. This publication is for deployers and administrators, and should be read first.

Prerequisite publications: IBM Tivoli Network Manager IP Edition

To use the information in this publication effectively when dealing with an integrated installation of Netcool Configuration Manager, Network Manager, and Tivoli Netcool/OMNIbus, you must have some prerequisite knowledge, which you can obtain from the Network Manager documentation, especially the following publications:

- *IBM Tivoli Network Manager IP Edition Release Notes*

  Gives important and late-breaking information about IBM Tivoli Network Manager IP Edition. This publication is for deployers and administrators, and should be read first.

- *IBM Tivoli Network Manager Getting Started Guide*

  Describes how to set up IBM Tivoli Network Manager IP Edition after you have installed the product. This guide describes how to start the product, make sure it is running correctly, and discover the network. Getting a good network discovery is central to using Network Manager IP Edition successfully. This guide describes how to configure and monitor a first discovery, verify the results of the discovery, configure a production discovery, and how to keep the network topology up to date. Once you have an up-to-date network topology, this guide describes how to make the network topology available to Network Operators, and how to monitor the network. The essential tasks are covered in this short guide, with references to the more detailed, optional, or advanced tasks and reference material in the rest of the documentation set.

- *IBM Tivoli Network Manager IP Edition Product Overview*

  Gives an overview of IBM Tivoli Network Manager IP Edition. It describes the product architecture, components and functionality. This publication is for anyone interested in IBM Tivoli Network Manager IP Edition.

- *IBM Tivoli Network Manager IP Edition Installation and Configuration Guide*

  Describes how to install IBM Tivoli Network Manager IP Edition. It also describes necessary and optional post-installation configuration tasks. This publication is for administrators who need to install and set up IBM Tivoli Network Manager IP Edition.

- *IBM Tivoli Network Manager IP Edition Administration Guide*

Describes administration tasks for IBM Tivoli Network Manager IP Edition, such as how to administer processes, query databases and start and stop the product. This publication is for administrators who are responsible for the maintenance and availability of IBM Tivoli Network Manager IP Edition.

- *IBM Tivoli Network Manager IP Edition Discovery Guide*

  Describes how to use IBM Tivoli Network Manager IP Edition to discover your network. This publication is for administrators who are responsible for configuring and running network discovery.

- *IBM Tivoli Network Manager IP Edition Event Management Guide*

  Describes how to use IBM Tivoli Network Manager IP Edition to poll network devices, to configure the enrichment of events from network devices, and to manage plug-ins to the Tivoli Netcool/OMNIbus Event Gateway, including configuration of the RCA plug-in for root-cause analysis purposes. This publication is for administrators who are responsible for configuring and running network polling, event enrichment, root-cause analysis, and Event Gateway plug-ins.

- *IBM Tivoli Network Manager IP Edition Network Troubleshooting Guide*

  Describes how to use IBM Tivoli Network Manager IP Edition to troubleshoot network problems identified by the product. This publication is for network operators who are responsible for identifying or resolving network problems.

- *IBM Tivoli Network Manager IP Edition Network Visualization Setup Guide*

  Describes how to configure the IBM Tivoli Network Manager IP Edition network visualization tools to give your network operators a customized working environment. This publication is for product administrators or team leaders who are responsible for facilitating the work of network operators.

- *IBM Tivoli Network Manager IP Edition Management Database Reference*

  Describes the schemas of the component databases in IBM Tivoli Network Manager IP Edition. This publication is for advanced users who need to query the component databases directly.

- *IBM Tivoli Network Manager IP Edition Topology Database Reference*

  Describes the schemas of the database used for storing topology data in IBM Tivoli Network Manager IP Edition. This publication is for advanced users who need to query the topology database directly.

- *IBM Tivoli Network Manager IP Edition Language Reference*

  Describes the system languages used by IBM Tivoli Network Manager IP Edition, such as the Stitcher language, and the Object Query Language. This publication is for advanced users who need to customize the operation of IBM Tivoli Network Manager IP Edition.

- *IBM Tivoli Network Manager IP Edition Perl API Guide*

  Describes the Perl modules that allow developers to write custom applications that interact with the IBM Tivoli Network Manager IP Edition. Examples of custom applications that developers can write include Polling and Discovery Agents. This publication is for advanced Perl developers who need to write such custom applications.

- *IBM Tivoli Monitoring for Tivoli Network Manager IP User's Guide*

  Provides information about installing and using IBM Tivoli Monitoring for IBM Tivoli Network Manager IP Edition. This publication is for system administrators who install and use IBM Tivoli Monitoring for IBM Tivoli Network Manager IP Edition to monitor and manage IBM Tivoli Network Manager IP Edition resources.

Prerequisite publications: IBM Tivoli Netcool/OMNIbus

To use the information in this publication effectively when dealing with an integrated installation of Netcool Configuration Manager, Network Manager, and Tivoli Netcool/OMNIbus, you must have some prerequisite knowledge, which you can obtain from the Tivoli Netcool/OMNIbus documentation, especially the following publications:

- *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*

Includes installation and upgrade procedures for Tivoli Netcool/OMNIbus, and describes how to configure security and component communications. The publication also includes examples of Tivoli Netcool/OMNIbus architectures and describes how to implement them.

- *IBM Tivoli Netcool/OMNIbus User's Guide*

  Provides an overview of the desktop tools and describes the operator tasks related to event management using these tools.

- *IBM Tivoli Netcool/OMNIbus Administration Guide*

  Describes how to perform administrative tasks using the Tivoli Netcool/OMNIbus Administrator GUI, command-line tools, and process control. The publication also contains descriptions and examples of ObjectServer SQL syntax and automations.

- *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*

  Contains introductory and reference information about probes and gateways, including probe rules file syntax and gateway commands.

- *IBM Tivoli Netcool/OMNIbus Web GUI Administration and User's Guide*

  Describes how to perform administrative and event visualization tasks using the Tivoli Netcool/OMNIbus Web GUI.

Accessing terminology online

The IBM Terminology website consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology website at the following Web address:

http://www.ibm.com/software/globalization/terminology

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Information Center website at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File** > **Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

Ordering publications

You can order many Tivoli publications online at the following website:

http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, contact your software account representative to order Tivoli publications. To locate the telephone number of your local representative, perform the following steps:

1. Go to the following website:

   http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss
2. Select your country from the list and click **Go**. The **Welcome to the IBM Publications Center** page is displayed for your country.
3. On the left side of the page, click **About this site** to see an information page that includes the telephone number of your local representative.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully.

With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education website:

https://www.ibm.com/training/search?query=tivoli

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

Online
Go to the IBM Software Support site at http://www.ibm.com/software/support/probsub.html and follow the instructions.

IBM Support Assistant
The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to http://www.ibm.com/software/support/isa

Conventions used in this publication

This publication uses several conventions for special terms and actions and operating system-dependent commands and paths.

Typeface conventions

This publication uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:** and **Operating system considerations:**)
- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs)
- Words defined in text (example: a nonswitched line is called a *point-to-point* line)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data
- Variables and values you must provide: ... where *myname* represents....

Monospace

- Examples and code examples

- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

This publication uses the UNIX convention for specifying environment variables and for directory notation.

When using the Windows command line, replace **$***variable* with **%***variable***%** for environment variables, and replace each forward slash (**/**) with a backslash (****) in directory paths. For example, on UNIX systems, the $NCHOME environment variable specifies the directory where the Network Manager core components are installed. On Windows systems, the same environment variable is %NCHOME%. The names of environment variables are not always the same in the Windows and UNIX environments. For example, %TEMP% in Windows environments is equivalent to $TMPDIR in UNIX environments.

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

Chapter 1. User administration

Use this information to administer Netcool Configuration Manager users.

Related reference

Change Compliance user names and passwords using the intellidenRmUser.sh script
Use the CLI to change user names and passwords.

About users and groups

ITNCM - Base access is managed using the Account Management Web Interface. Account Management consists of 'Groups' and 'Users', which are used to provide access privileges. This is an important security feature, because you can limit the ability of users and groups to perform certain actions by assigning rights and permissions. User IDs and passwords needed for system logins are created through the Account Management web interface.

There are three pre-defined default users and groups in ITNCM - Base: the administrator, operator and observer. Each of these default users and groups have different group permissions associated with them.

- **administrator** - All group activities are assigned to the administrator.
- **operator** - The following group activities are assigned to the operator: Execute Configuration Synchronization, Execute Configuration Change, Execute Direct Commands, Execute Import, View All Work, IDT Allow Manual Login, IDT enable Mode, IDT Access, View Native Commands, Apply Native Command Sets and Execute Compliance Policy.
- **observer** - The following group activities are assigned to the observer: IDT Allow Manual Login, IDT Access and View Native Commands.

User accounts must be created for access to ITNCM - Base, and group membership enhances the level of functionality available to users. Users' authorization for access, scope and functionality are all determined by the groups to which they belong.

In order to add groups and users, you need to have the appropriate permissions, the Manage Accounts activity, as well as Modify rights for the realm in which you are adding the group. To add security sets to a group, you also need Add rights for resources in that realm. A user's rights to realms and resources within realms are inherited from any group to which they belong. If a user belongs to several groups with different levels of security, the most lenient security settings apply.

Setting up administrator groups

ITNCM - Base is installed with a single super user account designed to be used solely for creating the administration groups. This system-installed group/user account cannot be used to work with resources, and cannot be used to delete or modify the group/user. A typical workflow is to use this user account to establish one or more administrator groups. These administrator groups can then be used to set up the remaining user accounts.

The following steps outline how to set up an administrator group.

1. Using the ITNCM - Base superuser account, create a group called Admin.
2. Assign the Admin group the appropriate permissions, the Manage Accounts activity, as well as Modify rights for the realm in which the Admin users will be adding groups/users.
3. Using the Admin account, create groups and users with various levels of access.

Administering user groups

You can create and modify groups, and add or remove users.

You need the `Manage Accounts` permission activity in order to be able to create or modify users and groups. By default the Superuser account, which was specified during ITNCM - Base installation, has

`Manage Accounts` permissions. A user is unable to modify their own account. For more information on permission activities, see "Group permissions" on page 5.

Follow these steps to work with groups:

1. Log into the Netcool Configuration Manager web interface.
2. Click **Account Management** to open the **Account Administration** GUI.
   A list of all groups is displayed on the left side of the window.
3. To remove a group, click on the name of the group and click **Remove**.

   Note: Before removing a group, assign the members of that group to a new group.
4. To create or edit a group, select the **New** icon next to Groups, or click the name of an existing group.
   Enter or update the group information:
   a) The **New Group** or **Modify Group** window is displayed.
   b) In the **General** tab, enter a Group Name (53 characters maximum) and Description (256 characters maximum) to identify the new group.
   c) Click the **Activities** tab. The Activities listing shows all available access privileges. Using the arrow keys in the middle of the screen, transfer the activities required into the "Authorized To" listing. For further information on permission activities, see "Group permissions" on page 5.
   d) Click the **Users** tab. Use the arrow keys in the middle of the screen, to ensure that all users who require membership to this group are moved to the "Assigned To" listing.
   e) Click the **Workflow** tab.
   f) From the **Realm** field, select the correct realm level for the new group. Make sure to select a realm at the same level, or higher than any realm/resources that have been given security rights in the **Security** tab.
   g) From the **Policy Set list**, select a policy set from the drop down menu. The selected policy affects the number of approvals required for configuration changes submitted by this group. A Policy Set of 0 does not require any approvals. Policy Sets of 1, 2, and 3 require 1, 2, and 3 approvals respectively.
   h) Click the **Security** tab. The **Realm**, **Resource**, and **Content Security** options are displayed.
   i) The **Realm** security options give you the ability to specify whether a group can view the realm, add subrealms to the realm, change the name of the realm, or delete the realm. Permissions assigned to a realm are inherited by its sub-realms. The following table explains the **Realm** security options.

Realm security option	Description
View	Allows users in the group to view the realm.
Add	Allows users in the group to add sub-realms under the realm.
Modify	Allows users in the group to modify/rename the realm. If the group has been granted "Manage Accounts" rights, you must also select this box to enable users in the group to create new groups in this realm.
Delete	Allows users in the group to delete the realm.
All	Selects all of the above rights for this realm. **Note:** In order to move a realm, that user must have Delete privileges in the realm being moved and Add privileges in the realm to which the sub-realm is being moved.

   j) Click the Resource link above the current form. The Resource Security form is displayed. Resource security determines if group members can work with resources (network resources, security sets,

command sets) within a given realm. For each realm, you can specify that the group can view resources, add resources, modify resources, delete resources, or apply command sets to resources. Assign the group rights to resources in each realm, using the descriptions in the following table. Any right that you assign to a realm is inherited by each sub-realm, but can be overridden. The following table explains the Resource security options.

Resource security option	Description
View	Allows users in the group to view all network resources, configurations, command sets, and security sets within this realm.
Add	Allows users in the group to create network resources in this realm and to import configurations into those resources. Users can also add command sets and security sets to this realm, and can associate security sets with groups.
Modify	Allows users in the group to change current or draft configurations, command sets, and security sets within this realm. Users can also rename current or draft configurations, network resources, command sets, and security sets.
Delete	Allows users in the group to delete resources, draft or versioned configurations, security sets, and command sets.
Execute	Allows users in the group to apply command sets to resources in this realm.
All	Automatically selects all rights for this realm.

k) Click the **Content** link above the form. The **Content Security** form is displayed. Content security determines if group members can work with command filtering within a given realm. Choose a realm and then a security set from the drop down lists for each realm to which you wish to apply a security set. Next, choose a command filter that you wish to apply to the realm you have specified.

Note: Both the security sets and the command filters must be created via the thick client before they are applied to the user group. Use the **Add** button to assign each Realm/Security Set/Command Filter relationship.

5. Click the **Save** or **Create** button.

Administering users

You can create, modify, and remove users by using the Account Management Web interface.

You need the `Manage Accounts` permission activity in order to be able to create or modify users and groups. By default the Superuser account, which was specified during ITNCM - Base installation, has `Manage Accounts` permissions. A user is unable to modify their own account. For more information on permission activities, see “Group permissions” on page 5.

Follow these steps to work with users:

1. Log into the Netcool Configuration Manager web interface.
2. Click **Account Management** to open the **Account Administration** GUI.

   A list of all groups is displayed on the left side of the window. A list of all users is displayed under the list of groups.
3. To create a new user from a remote repository such as LDAP, complete the following steps:
   a) Add the users in the remote repository.
   b) In the remote repository, add the users to the `IntellidenUser` group, and if they are an administrative user, also add them to the `IntellidenAdminUser` group.
4. To create or update a local user:

a) Click the name of an existing user or click **New** next to the **Users** list.

The **New User setup** screen or **Modify User** screen is displayed. Ensure that the **General** tab is selected.

b) Enter the details for the user. Mandatory fields are marked with an asterisk.

Field	Description
User Name	Enter a name (64 characters maximum) for the new user. The following words are reserved and cannot be used as user names: `ITNCM - Base`, `ftp`, `system`, `guest`, and `everyone` .
Remote User	This option is used only to support legacy features.
Two Factor User	This option is used only to support legacy features.
Password fields	Enter the password (64 characters maximum) for the new user, and then enter it again for verification. The password field is case-sensitive. If the user is a remote user, these fields are disabled.
First Name	Enter the first name of the new user. **Note:** The combined character length for the First Name, Middle Initial and Last Name fields must not exceed 64 characters. An error warning is received if the total character length for these fields exceeds 64 characters.
Middle Initial	Enter the middle initial of the new user.
Last Name	Enter the last name of the new user.
E-mail	Enter the e-mail address (100 characters maximum) for the new user.
Telephone #	Enter the telephone number (32 characters maximum) for the new user.
Identification	Enter the identification (32 characters maximum) for the new user.
Inactive	To inactivate a local user, check the **Inactive** box. To reactivate a local inactive user, uncheck the **Inactive** box. Inactivated users are marked with a red X icon in the user list. To inactivate a user from a remote user repository such as LDAP, remove them from the `IntellidenUser` and `IntellidenAdmin` groups. To reactivate a user from a remote user repository such as LDAP, add them to the `IntellidenUser` or `IntellidenAdmin`, or both, groups.

5. To remove a user, click the **Remove** button.

   User removals from ITNCM - Base are checked for dependencies; if the user has any work pending you might be unable to delete them. System users are marked with a red S icon in the user list and cannot be removed.

6. To change the group membership of users, select the **Groups** tab. The **Groups** selection box shows all groups that are available on that Netcool Configuration Manager - Base server. Using the arrows in the middle of the screen, move the groups required into the **Member Of** listing. A user must be a member of at least one group in order to gain access to Netcool Configuration Manager - Base.
7. Select **Save** or **Create** to accept the changes.

Group permissions

Group permissions (known as activities) are defined in the profile for each group. These functional access rights allow the users in the group to perform a number of predefined activities.

Activities

The following table identifies the valid permission activities along with a description that summarizes the allowable activity.

Permission activity	Description
Apply Native Command Sets	This activity is required to permit the users in a group to apply Native Command Sets. However, a user will still be allowed to apply a modelled Command Set without this activity.
Apply Wizard Preferences	This activity provides the users in a group with the ability to make changes to the UoW Submission wizard settings in User Preferences.
Execute Compliance Policy	This activity is required to be able to execute compliance policy.
Execute Configuration Change	Permits the users in a group to apply Configuration UOW.
Execute Configuration Synchronization	Permits the users in a group to perform configuration synchronizations on one or more resources.
Execute Direct Commands	Permits a user of the API to call a function on `irm\ResourceManagerSB` to submit a command directly to a network resource. This permission is also required to create native command sets.
Execute Imports	Permits the users in a group to import new resources into ITNCM - Base. Users can also make changes to a configuration, but they cannot submit the changes.
Housekeeping	Permits the users in a group to clean up completed UOWs and versioned configurations using the Work Housekeeping and Configuration Housekeeping utilities.
IDT Access	This activity provides access to IDT through the Tools menu. IDT Access is only required for login to a device via IDT. The user does not require any activities to view their own logs. They do however require IDT Administration to view other users' logs.
IDT Administration	Allows users to view active sessions and device logs for all other users.
IDT Allow Auto Login	Allows groups with this activity to use automatic login for a device. Must also have IDT Access permission.
IDT Allow Manual Login	Allows groups with this activity to use manual login only to gain access to the device. Must also have IDT Access permission.
IDT Enable Mode	Allows IDT to gain access to Enable Mode on a device if a device script is used. Either Auto Login or Manual Login mode must also be chosen.

Permission activity	Description
Manage Accounts	Permits the users in a group to add, modify, and delete users and groups from the system. Modifications can include changes to the data scope and changes in permissions of any group.
Manage Archive	Permits the users in a group to run the Archive housekeeping utility in icosutil from the command line.
Manage Compliance Policy	Permits the users in a group to create, edit, and delete compliance entities.
Manage Policy Remedial Work	Permits the users in a group to approve remedial work in the compliance remedial queue.
Manage System	Permits the users in a group to pause or restart ITNCM - Base. Users can also denote a realm as a system realm.
Manage Work	Permits the users in a group to approve, reject, or dequeue/cancel any UOWs submitted through the user interface or API. Users can dequeue their own UOW without "Manage Work" permissions. However, in order to dequeue another user's UOW "Manage Work" rights must be granted. If this permission is disabled, the user cannot override the default Pre-Emptive Compliance settings for the Hother three Pre-Emptive activities.
Pre-Emptive Compliance(Block on Failure)	Will block an Apply Commandset or Submit Config from completing if any Compliance failure exists against the projected config.
Pre-Emptive Compliance(Block on new failure)	Will block an Apply Commandset or Submit Config from completing if any compliance failure exists against the projected configuration and where the current configuration is compliant. Basically, the changes being applied are causing the compliance failure rather than any pre-existing failure.
Pre-Emptive Compliance(Report Only)	Will only report status of pre-compliance in the audit log. Changes to the device will still be made.
PrintAndSave Configuration	Permits the users in a group to do print, save and export to file operations in Netcool Configuration Manager.
Service Template Management	Allows users in the group (typically NSM service designers) to manage NSM service templates by using the `nsmadmin.sh` utility. NSM service templates allow for the easy and repetitive execution of Netcool Configuration Manager command sets in an ordered and controlled manner. NSM service templates also allow the execution of Netcool Configuration Manager device synchronizations and extractions.
Service Management	Allows users in the group (typically NSM client users) to use the NSM REST API URIs to POST services and GET information about services, service templates, devices, and realms from Netcool Configuration Manager.
OS Upgrade	Permits the users in a group to run OS Upgrades.

Permission activity	Description
View All Work	Permits the users in a group to see all the work that has been submitted on the system by all users. If the user does not have this activity, they shall only have the ability to view work from other users with whom they share a group.
View Archive	Permits the users in a group to view the Archive Manager in the user interface.
View Native Commands	Permits the users in a group to view Native Commands on a network resource.
View System	Permits the users in a group to view the Systems Manager in the user interface.

Chapter 2. System administration

Use this information to administer the Netcool Configuration Manager system. You can modify the reseries and system properties files, enable automated events to trigger compliance processes, change worker server pool sizes, and modify the LPTA timeout value.

Modifying system properties

The default system property value can be modified by administration users with the appropriate permissions.

To view system properties, a user must be in a group with the View System activity. To view and modify system properties, a user must be in a group with the Manage System activity.

1. To access System Properties from the Systems Manager, select **Tools** > **System Properties**.
   The System Properties screen is displayed and shows the following information for each property:
   - Property name
   - Short description of the property
   - Field type
   - Current value associated with the property
2. Select the system property to be modified from the upper pane of the **System Properties** screen.
   The current value held in the particular property chosen populates into the **Property Edit** pane in the bottom half of the screen.
3. Modify the property value as required, and then click **Update** to apply the changes.
4. Use the following table for more information on the system properties that you can modify.

Property name	Description
Activate Device Type Validation on Command Sets	Validate the Device type (T of VTMOS) when applying a command set. Default setting does not validate device type.
Allow Password Save	If selected, this populates the check box on the Login GUI for a "Password Save". Is False by default.
Apply CommandSet - allow stale config	Overrides that allow Command Set to be executed against a stale config.
Apply NativeCommandSet - allow stale config	Overrides that allow a Native Command Set to be executed against a stale config.
Apply Search Set - allow stale config	Overrides that allow a Search Set to be executed against a stale config.
Approve own work	If this is set to true, the user is allowed to approve their own work.
Compliance Policy Event Registration	Allows the administrator to turn off the Compliance Policy Event, so it is not written to the database. Compliance Policy Events are all events related to ITNCM - Compliance.
Config Editor Title	Changes the name of the Configuration Editor title bar.

Property name	Description
Configuration Change - allow stale config (Draft)	Overrides that allow a configuration change to be executed against a stale config (draft).
Configuration Change - allow stale config (Versioned)	Overrides that allow a configuration change to be executed against a stale config (versioned).
Connect on Driver Update	If this is set to true, ITNCM - Base will connect to the device during the driver update to confirm the VTMOS.
ConnectRetrydelaySeconds	Number of seconds after which a failed connection will retry.
ConnectTimeoutSeconds	Number of seconds after which a failed connection will timeout.
Custom Resource Browser label 1-10 **Note:** Custom labels are only supported for use in the Netcool Configuration Manager - BaseGUI, and not the Netcool Configuration Manager - Compliance (or any other) GUIs.	Additional labels for resources that a user can define when creating a new resource. Restart ITNCM - Base client for changes to take effect. **Tip:** The 'type' setting allows the value to be requested from a checkbox rather than a textbox.
Custom Resource Browser label state 1-10	User can choose if the Custom Resource label should be: Not Visible, API, Optional or Mandatory. • **Optional** means that when a device is created the user does not have to supply a value for the label. • **Mandatory** means that the user must supply a value. • **Not visible** means that the user cannot see the label. • **API** means that a value can only be created or modified by using the Netcool Configuration Manager Java API. Restart ITNCM - Base client for changes to take effect.
Custom UOW label 1-3	Additional labels for UOWs. Restart ITNCM - Base client for changes to take effect.
Custom UOW label state 1-3	User can choose if the Custom UOW label should be: Not Visible, Optional or Mandatory. Restart ITNCM - Base client for changes to take effect.
Default Device Enable Password	Default Enable password is used for device login. Restart ITNCM - Base for changes to take effect.
Default Device Password	Default Password is used for device login. Restart ITNCM - Base for changes to take effect.
Default Device Username	Default Username is used for device login. Restart ITNCM - Base for changes to take effect.
Error Word Delimiter	Regex used to look for errors in device responses.

Property name	Description
Event Polling Interval	Interval between polling for new JMS messages (for example, UOW changes).
FIPS Operating Mode	Indicates that the server is configured with FIPS compliant ciphers. (Read only)
Force Password Change	Forces the user to change password when they first login after creation or modification of the account. False by default.
FTP Password	Password used for accessing the FTP server. Default is that specified during installation.
FTP username	Username used for accessing the FTP server. Default is that specified during installation.
GUI Inactivity Timeout State	Activates the GUI inactivity timer.
GUI Inactivity Timeout Value	Inactivity time - after which the GUI will exit.
GUI Inactivity Warning Value	Time before exiting that the GUI will display a warning.
GUI Name Suffix	Sets the GUI suffix.
GUI Title	Changes the name of the application on the main title.
Hierarchial Account Security Mode	Show group hierarchies during account management. Default is False.
IDT Allow connections from Presentation	Allow IDT to connect to devices from Presentation server.
IDT Buffer Size	Maximum size of scrollback buffer in IDT Terminal. This can be used to configure the number of lines which are viewed when using the scroll function.
IDT Connection View	User may choose the view they want to see when connecting to a device. For example, they may see a logon script, an animation, a combination of the two, or nothing.
IDT Default UOW Description	This is the default description that is used when IDT submits a UOW.
IDT Device Output Limit	Limits number of characters logged for any device output. A limit of zero will return all output.
IDT Display UOW Confirmation	Displays a confirmation and UOW ID when the UOW has been submitted successfully.
IDT Inactivity Timeout Period	Number of minutes a session will stay connected during user inactivity. The default is 15 minutes. However, any timeout set on the device will override this property.

Property name	Description
IDT Regex Match String	Regex commands may be entered and searched upon before forcing synchronization. This is only valid if the "IDT Synchronization Disconnect Action" is set to "Regex Match Synchronization".
IDT Synchronisation Disconnect Action	The Synchronization action to perform once a user has disconnected from a device can be chosen. The options are Prompt Synchronization, Force synchronization or use Regex match synchronization.
IDT Terminal Throttle	This configures the maximum number of terminal sessions that each client can be running at any given time.
IDT UOW Conflict Connection	Action to take when connecting to a device.
IDT Use Main Server For Connection	If you want device connections to be facilitated by the master presentation server, set this option to "True" on all presentation servers.
IDT Use Default Device Credentials	Configures auto login IDT credentials. By default it is set to true, which means IDT will attempt to connect using the default device credentials. If set to false, IDT will not use the default device credentials to connect unless no other username/password enable prompts are retrieved from a relevant authentication GR. The default device credentials can be configured using the properties listed earlier in this table: Default Device Enable Password, Default Device Password and Default Device Username.
Instance Identifier	A unique name should be provided for this particular instance of NCM, for example, ITNCM.
Internal Housekeeping - Interval	Frequency that Internal Housekeeping is performed.
Internal Housekeeping - JMS Message Keep Time	Configure the number of days to store JMS messages.
Internal Housekeeping - UOW Information Keep Time	Configure the number of days to store internal UOW information.
ITNCM Help URL	Default URL to access ITNCM help.
ITNCM WebStart Server	The fully qualified hostname or the IP Address of the server that serves the JNLP launch file for WebStart client applications. The default value is `localhost`. To revert to the default value, enter a single space character and save. If you enter a server address here, it must not include the transport, such as `http://` or `https://`. The server address must not contain any port number. An example of a valid server address is: `ncmwebserver.mycompany.com`

Property name	Description
	An example of a valid IP address is: `198.12.0.1`
Logon Message	Configure the content of a Logon Message. This appears when the user has entered ITNCM - Base login credentials, and selected the Login button; appears before ITNCM - Base is invoked. Message will be displayed only when the Logon Message Display is set to True.
Logon Message Display	If True is selected, the properties specified in the Logon Message and Logon Message Title will be displayed. False by default.
Logon Message Title	Configure a title for the Logon Message.
Maximum Client Memory	Maximum Java Heap Setting for WebStart client.
MaxResponseTimeoutSeconds	Maximum time for device to send back complete response.
Memory Manager - Best Effort Allocation	IBM Tivoli Use Only*
Memory Manager - Debug Enabled	IBM Tivoli Use Only*
Memory Manager - Default Driver Memory	IBM Tivoli Use Only*
Memory Manager - Default Driver Memory (64 bit)	IBM Tivoli Use Only*
Memory Manager - Default Task Memory	IBM Tivoli Use Only*
Memory Manager - Default Task Memory (64 bit)	IBM Tivoli Use Only*
Memory Manager - Driver Memory Scale Factor	IBM Tivoli Use Only*
Memory Manager - Driver Memory Scale Factor (64 bit)	IBM Tivoli Use Only*
Memory Manager - Enabled	IBM Tivoli Use Only*
Memory Manager - Maximum Wait Time	IBM Tivoli Use Only*
Memory Manager - Memory Reserved For Drivers	IBM Tivoli Use Only*
Memory Manager - Memory Reserved For Drivers (64 bit)	IBM Tivoli Use Only*

Property name	Description
Memory Manager - Memory Reserved For Reallocation	IBM Tivoli Use Only*
Memory Manager - Memory Reserved For Tasks	IBM Tivoli Use Only*
Memory Manager - Memory Reserved For Tasks (64 bit)	IBM Tivoli Use Only*
Memory Manager - Percent Free Memory Allocatable	IBM Tivoli Use Only*
Memory Manager - Schema Scale Factor	IBM Tivoli Use Only*
Memory Manager - Schema Scale Factor (64 bit)	IBM Tivoli Use Only*
Minimum Client Memory	Minimum Java Heap Setting for WebStart client. Default is 32MB.
Network Resource Event Registration	Allows the administrator to turn off the Network Resource Event, so it is not written to the database. Network Resource Events are those affecting network resources such as command sets being applied, configuration synchronization and import, and so forth.
Page Size for Native Commands	Number of lines per page for a Configuration's native source. Default is 10000. Please note any change to this setting will only take effect when the device is re-imported.
Resource Event Registration	Allows the administrator to turn off the Resource Event, so it is not written to the database. Resource Events are those affecting the movement of all resources in the ITNCM system. Create, move, rename and delete of resources are all covered by these events.
Show modelled hardware	When set to True, this property displays Modelled Hardware in the Hardware tab. When set to True, Native Hardware will be displayed.
Server Timeout	Number of seconds after which a failed connection to server will timeout.
SNMP Local Engine ID	Allows configuration of the SNMP engine ID
SNMP Trap Flood Prevention	The minimum time, in milliseconds (ms), that the system waits after sending a trap before sending another trap. For example, if this property is set to the value 3, the system will send no more than one trap every 3 ms.
SNMP Trap Recipients	Specifies a list of servers to which the SNMP trap messages are sent. The value should take the following form: hostname:port:optional community string, e.g.

Property name	Description
	192.168.20.138:162. If there are multiple SNMP trap recipients specified, they should be separated by commas.
SNMP Trap Retries	The maximum number of times the system will attempt to resend unacknowledged traps.
SNMP Trap Timeout	The maximum time, in milliseconds (ms), that the system waits for a trap acknowledgement before resending the data.
SocketConnectTimeoutSeconds	Timeout for connecting to the socket.
Synch from ITNCM - Base to Device - allow stale config	Determines whether a stale config can be synched to a device.
Task Lock Clean Up Interval -2	Every 2 minutes the lock clean-up code will run and poll the locks table for locks that should be removed. That is, locks for finished tasks, and locks for tasks not begun within a specified period.
Task TimeOut - 5	If a thread locks a device and does not set the task to execute within 5 minutes, the lock is cleared, and another thread will pick up the task.
tipserver url	URL of Integrated ITNM-TIP Server/Local TCR Server.
tip wizard - disaster recovery	Disaster recovery option for the Submit Configuration TIP wizard. When applying a versioned configuration if a device is unmanageable enabling disaster recovery will push the entire versioned configuration to the device using the native CLI commands.
tip wizard - execution order	Determines the execution order of command sets in the Apply Modelled Command Set and Apply Native Command Set TIP wizards. Options are Apply device at a time and Apply command set at a time.
tip wizard - FAILURE OPTION % ERRORS	Determines the total percentage of errors allowed before failure for a UOW submitted via an Apply Modelled Command Set or Apply Native Command Set TIP wizards. Only applicable if the 'TIP Wizard - Failure Option Type' system property has been set to 'Fail After Percentage Errors'.
tip wizard - failure option total errors	Determines the total number of errors allowed before failure for a UOW submitted via an Apply Modelled Command Set or Apply Native Command Set TIP wizards. Only applicable if the 'TIP Wizard - Failure Option Type' system property has been set to 'Fail After Total Errors'.
tip wizard - failure option type	Failure options when dealing with multiple devices and/or multiple command sets in the Apply Modelled Command Set and Apply Native Command Set TIP wizards. Options are Ignore All Errors, Fail After Total Errors, and Fail After Percentage Errors.

Property name	Description
tip wizard - pre-emptive compliance	Pre-Emptive Compliance options for Apply Modelled Command Set, Apply Native Command Set, and Submit Configuration TIP wizards. Options are No Pre-emptive Compliance, Report Compliance Failures Only, Block Configuration Change On New Compliance Failures, and Block Configuration Change On Any Compliance Failures.
tip wizard - rollback mode	Determines the rollback mode for the Apply Modelled Command Set and Apply Native Command Set TIP wizards. Options are None, Rollback Failed Network Resources, and Rollback All Command Sets.
tip wizard - rollback options	Determines the rollback options for the Apply Modelled Command Set and Apply Native Command Set TIP wizards. Options are No Rollback, Modelled Rollback, Reboot Device, and Modelled Rollback and Reboot Device.
tip wizard - rollback verification	Determines whether rollbacks should be verified for the Apply Modelled Command Set and Apply Native Command Set TIP wizards.
Welcome Message	Configure the content of a Welcome message. This appears when the user has successfully logged on to ITNCM - Base. Message will be displayed only when the Welcome Message Display is set to True.
Welcome Message Display	If True is selected, the properties specified in the Welcome Message and Welcome Message Title will be displayed. False by default.
Welcome Message Title	Configure a title for the Welcome Message as above.
Wizard * <screen name>	Visibility configuration for all screens involved in the UOW submit wizard.
Work Event Registration	Allows the administrator to turn off the Work Event, so it is not written to the database. Work Events are those which change the state of UOWs.
Worker Server Control Update Period	Controls how often the Worker Server Control State is updated. If changes are made to a Worker Server, this property dictates how often the updates shall run and therefore how quickly any modifications are applied. Default is 30 seconds.
Write Verified Changes	When set to True, it performs a final compare between the new running config and the old current config to show changes. It then writes the diffs out to the UOW log. The workflows affected are Apply Config (submit), Apply Command Set, Apply Native Command Set.
Default Device Loader Realm	This specifies the realm where devices from NCM shall be placed by default. The value should be in the following format: .

Property name	Description
	<Default device loader realm>/ *<ITNM Domain name>/* For example: `MyServer/domains`.

5. Optional: To revert the value to the original default value, click **Default** > **Update**.
6. Click **Close** to exit the **System Properties** window.

Modifying rseries.properties

The `rseries.properties` file contains properties that the system uses during runtime. You may need to edit some of these runtime properties depending on your system architecture and server configuration.

The `rseries.properties` file can be found in the following location: `/opt/IBM/tivoli/netcool/ncm/config/properties`

Use a text editor to modify the file.

The following table describes the configurable properties within `rseries.properties`:

Note: Properties marked INTERNAL/SUPPORT should only be changed by an IBM Level Two support engineer.

Property name	Description
AdminManager/ServerName	The unique name given to the server at install time and is shown under Systems Manager|Servers.
auditLogger/dbWriteQueueSize	The maximum number of unit of work (UoW) messages that can be in the internal queue for writing UoW messages to the database (INTERNAL/ SUPPORT).
Core/smtpServer	The smtp server entered at install time that will be used to send emails if notifications are turned on under User Preferences File |P references | Work Notifications.
Core/useCachedAuthCredential	If this property is set to true then ITNCM will cache the user name and password from authentication that was used to successfully log into the device in the database. The next time a UoW is executed against that device, the cached credentials will be tried first from the list of credentials in the autentication.
Core/workerServerUpdatePeriod	Sets the interval, in seconds, at which the worker refreshes its view of the Resource Browser in order for it to determine what work it can pick up. The default is 30 seconds.
deviceConstants/FTP_HOST	The ftp server that will be used to transfer configurations from devices if the streaming flag(s) are unselected in the resource access document of the device. Typically, this is the current server IP address/hostname.
HADRErrorCodes	The default setting is HADRErrorCodes=4498.

Property name	Description
	If this property is hashed out or an empty value is provided, Netcool Configuration Manager will try to reconnect to the database after a database connection failure has been detected.
HideInactiveUsers	A boolean parameter, which allows you to hide or show inactive users in the account manager. By default, the value is set to `false`. If it is set to false, all the users would be visible. If set to `true`, it will hide inactive users.
IDT/daemonPort	The port that the daemon listens on for connection requests from the webstart clients. Default set during install: 8104.
IDT/Hostname	Can be used to override the hostname of the presentation server used for IDT sessions if you do not want to use the default resolved hostname of the server. This setting is only used if IDT/useHostname is set to true. For example, if the hostname of the server is myhost@work.com and you set IDT/useHostname = TRUE, then the hostname used will be myhost@work.com. If you then set IDT/Hostname = test.com, the hostname used will be test.com.
IDT/mainserver	Either True or False. If this is set to True and the "IDT Use Main Server For Connection" system property is set to True as well, then IDT will connect through the presentation server that is marked as the main IDT server. If set to False, the connection will be made through the current presentation server.
IDT/standaloneListenPort	The port that the stand alone server listens on for control commands. Defaults to 8105.
IDT/useHostname	Either True or False. Controls whether IDT uses the hostname for the IDT daemon or the IP address. When starting up and shutting down, it creates an entry in the database in the IDT_DAEMONS showing the status of the daemon. IDT also records the IP address/hostname depending on what is selected in this property.
jpa/connectionURI	The connection uri to the database. For example: `jdbc:oracle:thin:@<`*`db ip address`*`>:<`*`port`*`>:<`*`oracle sid`*`>`
jpa/mediumMinIdle	The minimum number of idle connections to the database (INTERNAL/SUPPORT).

Property name	Description
	On startup the server creates the specified number of idle connections. The value can be tweaked depending on the number of worker servers and processes that you have set up on the database.
jpa/mediumMaxIdle	The maximum number of idle connections to the database (INTERNAL/ SUPPORT). The value can be tweaked depending on the number of worker servers and processes that you have set up on the database.
NMEntityMappingComponent/ securityComponent	Specifies the TLS version used to connect to Network Manager WebSphere. For example, if you change the Network Manager Websphere version to TLS version 1.2, set this property to TLSv1.2. The default is TLSv1.
NMEntityMappingComponent/baseURL	The connection method for the Network Manager server, its hostname, and port number. The connection method can be HTTP or HTTPS. Example: `https://myitnmserver.myorg.com:16311/`
NMEntityMappingComponent/uri	The URI for retrieving the devices from Network Manager. To return data from devices within a specific domain, include the domain name. For example: `ibm/console/nm_rest/topology/devices/domain/`*`domain_name`* To return all devices, use: `/ibm/console/nm_rest/topology/devices/all/`
NMEntityMappingComponent/uriProps	This parameter allows you to specify Network Manager REST API parameters for more specific selection of devices to import.
NMEntityMappingComponent/uriParam	Allows specification of particular devices. For instance, by listing the Network Manager entity IDs you want. Example: `param_entityIds=19,20`
NMEntityMappingComponent/user	The user name that is used to log on to Network Manager. By default, this is `itnmadmin`. The Network Manager username must have permissions perform the required functionality.
NMEntityMappingComponent/passwd	The password for this user.
NMEntityMappingComponent/period	The waiting time between imports or sync with Network Manager (in minutes).
NMEntityMappingComponent/delay	The waiting time before first import or sync with Network Manager (in minutes).

Property name	Description
NMEntityMappingComponent/ncmUser	The Configuration Manager user under which the import or sync is performed.
NMEntityMappingComponent/importRealm	The Configuration Manager realm that the devices imported from Network Manager are created in.
NMEntityMappingComponent/ maxDevicesPerRealm	The maximum number of devices which are imported into one realm. Set to `0` for no limit.
NMEntityMappingComponent/ deviceNameFilter	This parameter filters devices based on a device name. Any devices returned from Network Manager which match this setting are not imported into Configuration Manager.
NMEntityMappingComponent/ allowDuplicateDeviceForSeparateDomains	A boolean parameter, which determines whether the import can import duplicates. Set to `false` as default.

Enabling or disabling automatic validation

Enable or disable the automatic validation parameter setting by using the Options dialog. When enabled, the automatic validation parameter permits automated events to trigger compliance processes. When disabled, the automatic validation parameter does not permit automated events to trigger compliance processes.

The Options dialog is available within the Admin menu. The Options dialog provides the Automatic Validation Trigger setting used to enable or disable the automatic validation parameter. By default, the automatic validation parameter is set to Disabled. The Automatic Validation Trigger setting is a system wide setting.

Automated events are associated with changes in ITNCM - Base (that is, changes in realms, devices, command sets, and native command sets) that trigger compliance processes.

To enable or disable automatic validation, follow these steps.

1. From the Admin menu, display the Options dialog. The following table describes each of the fields in the Options dialog.

Option	Description
Option	**Description**
Automatic Validation Trigger Setting	Specifies the name of this dialog.
Automatic Validation Parameter	Specifies the parameter to be enabled or disabled.
Automatic Validation Trigger:	Specifies the drop down selection box to enable or disable the automatic validation parameter.
Cancel	Cancels the operation.
Apply	Applies the selection: either enable or disable the automatic validation parameter.

2. Using the drop down selection box adjacent to the Automatic Validation Trigger dialog item, select Enable or Disabled.
3. Click **Apply**.

Changing server pool sizes

The default worker server pool sizes can be configured to meet the system requirements.e

This procedure describes how to modify the default worker server pool sizes.

1. To modify the worker server pool sizes, right click on the worker server and choose **Change pool settings** from the pop-up menu.
2. The **Change Server Pool Sizes** screen is displayed. The max normal pool size can be configured to address the number of threads set on a worker server. This needs to be optimized based on device real estate, but generally a value between 20 and 40 threads is applicable. The range will be affected by the amount of heap available to the JVM and to the number of database connection available.
3. Click **OK** to save changes to the server pool size.

Modifying LTPA timeout

The default Lightweight Third-Party Authentication (LTPA) timeout value can be modified to meet requirements.

LTPA timeout value is a WebSphere Application Server global security setting. To change it, you require access to the WebSphere Application Server Integrated Solutions Console. Only an admin user can change this setting.

The LTPA timeout value default is 480 minutes (eight hours), after which the authentication token that enables the single-signon across Netcool Configuration Manager interfaces expires. This means, for example, that GUIs that have been open for over eight hours, or have been left open over night, are closed automatically. You can change this value to suit your requirements.

1. Access the Integrated Solutions Console on the server on which the DASH is installed. Use the following URL:
   `https://<server ip>:18101/ibm/console`
   where *server ip* is the IP address of the TIP server.
2. Log onto the console as the superuser.
3. Select **Global Security** under Security on the left hand side.
   The **Global Security** window is displayed.
4. In the Authentication section, click **LTPA**.
   The **Global Security>LTPA** window is displayed.
5. Adjust the value in the **LTPA timeout** text field as required.
6. Click **OK** to save your changes.

Chapter 3. Setting user preferences

Use this information to set user preferences for the Netcool Configuration Manager interfaces. Changed preferences take effect the next time you access the interface.

You access the **Preferences** window from the Netcool Configuration Manager GUI. You can set user preferences for the following interfaces:

Archive Manager
Select this option to make changes to Archive Manager including queue table, work logs, the display of detail dialogs, and refresh options

Configuration Editor
Select this option to select your list view options for roll-up lists.

These settings are stored in the user's home directory. If more than one person uses the same machine, they can have their own settings stored and used as long as each user logs into Netcool Configuration Manager with a unique account. Configuration Editor preferences will follow the user across machines if the user's home directory is shared across machines.

Detail Tabs
Select this option to configure the refresh options on the information tabs in the Queue Manager, Archive Manager and Resource Browser.

General Application
Select this option to choose general application settings such as the showing of confirmation dialogs.

Paging
Select this option to control the page sizes available for selection in the Paging panels.

Queue Manager
Select this option to make changes to Queue Manager including queue table, work logs, the display of detail dialogs, and refresh options.

Queue Manager settings are stored in the user's home directory. If more than one person uses the same machine, they can have their own settings stored and used as long as each user logs into Netcool Configuration Manager with a unique account. Queue Manager preferences will follow the user across machines if the user's home directory is shared across machines.

Resource Browser
Select this option to change views and refresh options within the resource browser.

Systems Manager
Select this option to change refresh and view options for the Systems Manager.

User Information
This preference enables you to enter user information including email, telephone, and other identification information.

User Information settings are stored on the server, which means that the client machine can be shared between users.

User Password
This preference enables you to change the user password.

Wizard Panels
This preference enables the user to remove steps from the UoW submission wizards.

Work Notifications
This preference enables you to change Work Notifications and who will be notified, and at what level.

Work Notifications settings are stored on the server, which means that the client machine can be shared between users.

Setting Archive Manager preferences

The Archive Manager preferences enable you to customize the default view and configure the default behavior of the Archive Manager.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **Archive Manager** on the navigation tree.

   The **Archive Manager Preferences** dialog is displayed.
3. Set the following preferences.

   Default Refresh
   Select the desired refresh rate. Regardless of what value you select, you can manually refresh the Queue Manager at any time.

   Column Resize Mode
   Select the option for how you want the columns to behave when you resize the queue manager table.

   No resizing (Scroll Horizontally)
   When you resize a column, all other columns stay the same size, and a scroll bar is added to the bottom.

   Resize the Next Column
   When you resize a column, only the next column changes to compensate.

   Resize Subsequent Columns
   When you resize a column, all the columns to the right change to compensate.

   Resize Last Column
   When you resize a column, only the last column is changed to compensate.

   Resize All Columns
   When you resize a column, all the other columns are changed to compensate.

   Refresh

   Automatically refresh UOW List when actions finish
   Select this check box to enable automatic refresh of the UOW list when actions complete.
4. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting Configuration Editor preferences

The Configuration Editor preferences enable you to customize the default view and configure the default behavior of the Archive Manager. You can speed up the Configuration Editor by specifying that it only show a certain number of lines at a time.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **Configuration Editor** on the navigation tree.

   The **Configuration Editor Preferences** dialog is displayed.
3. Set the following preferences.

   Number of lines to trigger list view
   Select the minimum number of list commands you want displayed in list mode.

   Number of lines shown in list view
   Select the number of lines you want shown, without having to scroll, when list commands are in list mode. This setting will not affect list commands having less commands than the number you entered in the previous field.
4. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting Detail tabs preferences

The Detail tabs preferences customize the refresh options of the tabs in the Queue Manager, Resource Browser and Archive Manager.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **Detail Tabs** on the navigation tree.

   The **Detail Tabs Preferences** dialog is displayed.
3. Set the following preferences.

   Refresh Tabs On Selection
   Select **Always** to ensure that the tab will be refreshed each time it is opened.

   Select **If Resource has Changed** to ensure that the tab will only be refreshed if the resource selected in the main Netcool Configuration Manager table has changed since the last time it was selected.

   Refresh Tabs After Action
   Select **Automatically refresh Tab Table when modify actions finish** to ensure that the tab table is refreshed after each action completes.
4. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting General Application preferences

General Application preferences enable you to control if you want to view verification dialogs when you close the application.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **General Application** on the navigation tree.

   The **General Application Preferences** dialog is displayed.
3. Select **Show confirmation when closing the application** to ensure that a confirmation dialog is displayed when you close the application.
4. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting Paging preferences

Paging preferences customize the page sizes that are available for selection in the paging panels in the Queue Manager, Archive Manager and Resource Browser.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **Paging** on the navigation tree.

   The **Paging Preferences** dialog is displayed.
3. Set the following preferences.

   Page sizes
   Add or remove entries from the page size drop-down list.

   Restriction: The system performs error checking when you add a new entry, ensuring that it is a whole number between 1 and 100,000.

   Refresh Options
   Select **Disable paging options during refresh** to improve refresh performance.
4. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting Queue Manager preferences

The Queue Manager preferences enable you to customize the default view and configure the default behavior of the Queue Manager.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **Queue Manager** on the navigation tree.

   The **Queue Manager Preferences** dialog is displayed.
3. Set the following Queue table preferences.

   Default Refresh
   Select the desired refresh rate. Regardless of what value you select, you can manually refresh the Queue Manager at any time.

   Column Resize Mode

   No resizing (Scroll Horizontally)
   When you resize a Queue Manager table column, all other columns stay the same size, and a scroll bar is added to the bottom.

   Resize the Next Column
   When you resize a column, only the next column changes to compensate.

   Resize Subsequent Columns
   When you resize a column, all the columns to the right change to compensate.

   Resize Last Column
   When you resize a column, only the last column is changed to compensate.

   Resize All Columns
   When you resize a column, all the other columns are changed to compensate.
4. Select **Automatically refresh Queue Table when modify actions finish** to enable automatic refresh of the queue table when actions complete.
5. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting Resource Browser preferences

Resource Browser preferences enable you to customize the default view and configure the default behavior of the Resource Browser.

The Resource Browser allows you to view and work with all types of resources used by Netcool Configuration Manager. You can specify settings for the Resource Browser that limit what you can view. For example, if you never work with a certain type of resource, you can remove that type of resources from your view.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **Resource Browser** on the navigation tree.

   The **Resource Browser Preferences** dialog is displayed.
3. Set the following Visibility preferences.

   Show Hidden Realms
   Select this option if you want hidden realms to be shown in the browser.

   Show System Realms
   Select this option if you want system realms to be shown in the browser. The default is to not show System realms.

   Restriction: You won't be able to change this option unless you have Manage System rights.
4. Set the Double Click Action preferences to determine the action to be performed on a resource when it is double-clicked. The default action saved here, is displayed in bold when a right click operation is made on the selected device in the Resource Browser.

Select one of the following options from the Actions drop-down list:

Note: When you double click on a device, the default option selected here is overridden based on the support level of the device. If the device does not support the default option chosen here, the action carried out on the device will be the first option in the list below that the device supports. For example, if you double click on a device with a Standard support level and the `View Configuration (Modelled)` action is chosen as the default, the action carried out on the device will be `View Configuration (Native)`.

View Configuration (Modelled)
Opens the Configuration Editor with the device's current configuration in read-only mode.

Edit Configuration (Modelled)
Opens the Configuration Editor with the device's current configuration in edit mode

View Configuration (Native)
Opens the Native Commands dialog with the device's current configuration.

IDT Manual Launch
Initiates a manual IDT session with the device.

5. Set the Columns preferences to determine how you want the columns to behave when you resize the Resource Manager table.
Select one of the following Column Resize Mode options:

No resizing (Scroll Horizontally)
When you resize a Resource Manager table column, all other columns stay the same size, and a scroll bar is added to the bottom.

Resize the Next Column
When you resize a column, only the next column changes to compensate.

Resize Subsequent Columns
When you resize a column, all the columns to the right change to compensate.

Resize Last Column
When you resize a column, only the last column is changed to compensate.

Resize All Columns
When you resize a column, all the other columns are changed to compensate.

6. Select **Show Update Flags** to display the red and orange indicator flags which appear if device drivers need to be updated.
7. Select **Automatically refresh Resource List when modify actions finish** to enable automatic refresh of the resource list when actions complete.
8. Under Show the Following Resource Types, select the resources to be displayed in the Resource Browser.

   Tip: Excluded resource types can still be accessed through the search function.
9. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting Systems Manager preferences

Systems Manager preferences enable you to customize the default view and configure the default behavior of the Systems Manager.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **Systems Manager** on the navigation tree.

   The **Systems Manager Preferences** dialog is displayed.
3. Set the following Visibility preferences.

Show Active Servers
Select this option if you want to display all active servers on your network.

Show Inactive Servers
Select this option if you want to display all **inactive** servers on your network.

4. Set the Columns preferences to determine how you want the columns to behave when you resize the Systems Manager table.
   Select one of the following Column Resize Mode options:

   No resizing (Scroll Horizontally)
   When you resize a Systems Manager table column, all other columns stay the same size, and a scroll bar is added to the bottom.

   Resize the Next Column
   When you resize a column, only the next column changes to compensate.

   Resize Subsequent Columns
   When you resize a column, all the columns to the right change to compensate.

   Resize Last Column
   When you resize a column, only the last column is changed to compensate.

   Resize All Columns
   When you resize a column, all the other columns are changed to compensate.
5. Select **Automatically refresh Systems List when modify actions finish** to enable automatic refresh of the systems list when actions complete.
6. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting user information

Edit your personal information using the User Information dialog.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **User Information** on the navigation tree.
   The **User Information** dialog is displayed.
3. Type the following information:

   Name Fields
   Edit your name as necessary. Any changes you make will not affect the login name that is displayed for any work you submit.

   E-Mail Address
   If you are going to sign up for work notifications, you must enter a valid email address in this field.

   Telephone Number
   Enter a phone number for contact purposes.

   Note: This field is not used by Netcool Configuration Manager.

   Identification
   Use this field for any other identification that your company requires.

   Note: This field is not used by Netcool Configuration Manager.

   Group Membership
   This field shows the groups of which you are a member.

   Time Zone
   This field shows the time zone where your member status resides.

   Tip: For display purposes, it is recommended that you set the system time zone appropriate to your location. When set, all times shown in the application will be converted to your timezone with the exception of the times shown in the UOW log, which will remain in GMT. When the timezone is set, it only applies against the username that was used to set the timezone.
4. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting user password

You can change the Netcool Configuration Manager password defined by your system administrator.

Ensure you are aware of the minimum password requirements you must adhere to before changing your password.

If single sign-on (SSO) has been enabled between Netcool Configuration Manager and Network Manager, you change the user password from the DASH console of the Netcool Configuration Manager Presentation Server, rather than the native Netcool Configuration Manager GUI.

This procedure only changes the user password that you use to logon to the Netcool Configuration Manager GUI, but not the database or Reporting user passwords. For information on how to change these, see the What to do next section.

Changing the Netcool Configuration Manager user password if single sign-on **has not** been enabled:

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **User Password** on the navigation tree.

   The **Password Change** dialog is displayed.
3. Type the following information:

   Current Password
   Type your current password.

   New Password
   Type the new password you want to assign to yourself.

   Confirm Password
   Type the new password again.
4. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Changing the Netcool Configuration Manager user password if single sign-on **has** been enabled:

5. Logon to the DASH console of the Netcool Configuration Manager Presentation Server.

   Note: The default logon is `http://host:port/ibm/console`
6. Change the password using the DASH interface.

To change the passwords for the reporting user and the database user, follow these steps:

To change a reporting user password

1. Logon to the DASH Reporting Server as the user for whom you want to change the password.
2. Select the user icon, then click **Change Password**.
3. Change the password, and save your changes.

To change the database user password

Use the `/opt/IBM/tivoli/netcool/ncm/bin/icosadmin` script with ChangeDbPassword as a flag, and the following options defined:

-u (database username)

-p (current database password)

-n (new database password)

For example:

```
icosadmin ChangeDbPassword -u <db_username> -p <old_db_passwd>  -n
    <new_db_passwd>
```

Note: If the Netcool Configuration Manager database password is changed, this impacts Reports. Therefore, you must change the Netcool Configuration Manager datasource password in the Reports as described here:

1. From Common Reporting launch Administration.
2. Select **Data Source Connections** > **ITNCM** > **View Signons** > **Set Properties** > **Signon** > **Edit the Signon**, and update the stored database password.

Setting Wizard Panels preferences

Wizard Panels preferences customize the unit of work (UOW) submission wizard. You can hide specific screens from the display sequence, and use the default value for those screens instead.

Ensure you are aware of the minimum password requirements you must adhere to before changing your password.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **Wizard Panels** on the navigation tree.

   The **Wizard Preferences** dialog is displayed. All possible wizard screens that are displayed during any type of UoW submission are listed. The default value set for each screen in System Properties is indicated in parenthesis after the screen name.
3. For each of the wizard screens listed, you can apply from the following options:

   Default
   Select this option to ensure the wizard screen applies the default value as specified in System Properties

   Hidden
   Select this option to ensure that this wizard screen is not displayed as part of the UOW Submission wizard screen sequence.

   Visible
   Select this option to ensure that this wizard screen is displayed as part of the UOW Submission wizard screen sequence.
4. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

Setting Work Notifications preferences

Work Notifications allow you keep track of a UOW's progress without having to log in. Instead, you receive emails when the work changes states.

In order to receive work notifications, you must ensure that you have a valid email address entered in the system.

You can be notified when a UOW changes to one or more specific state. UOWs can be those submitted by you, or by a member of your group, or they can be UOWs for which you are an approver.

1. To access the **Preferences** window, click **User Preferences**, or click **File** > **Preferences** from the main dialog.
2. Select **Work Notifications** on the navigation tree.

   The **Work Notifications** dialog is displayed and all work states are listed.
3. columns next to the work states for each association (user, group, approver).
4. For each of the listed UOW states, select one or more of the following options:

   Note: To be notified of all state changes, select **All**.

   User
   Select the **User** checkbox against a specific state in order to be notified when a UOW you submitted enters that state.

   Group
   Select the **Group** checkbox against a specific state in order to be notified when a UOW that was submitted by someone in your user group enters that state.

Approver

Select the **Approver** checkbox against a specific state in order to be notified when a UOW for which you are an approver enters that state.

Restriction: Selecting the **Approver** checkbox against a specific state will only have an effect if you have 'Manager Work' permissions.

5. Click **Apply** to save the settings, or click **OK** to save the settings and exit the **Preferences** window.

You will start to receive emails as soon as a UOW enters a state for which you signed up.

Chapter 4. Device communication

Use this information about Netcool Configuration Manager to administer communication for your devices.

About device communication

ITNCM - Base uses a number of objects that affect communication when interacting with devices. These objects — Resource Access Docs (RAD), Authentication, File Transfer, Device scripts, and ITNCM - Base Source Address Definition — interact with ITNCM - Base code to control communication with devices.

The following sections describe how to format the information associated with these objects to manage devices in ITNCM - Base.

Device authentication

When Netcool Configuration Manager sets up the communication between itself and a device, it builds a list of credentials to use when authenticating.

The application starts at the first credential set and continues to the last set or until success is achieved. The recommended method of supplying usernames and passwords is the Authentication GR. A default Authentication GR is created in the Resource Browser at install time. The end user that is submitting the UOW can enter the username, password, and enable-password.

If the user has set the `Core/useCachedAuthCredential` in `rseries.properties` to TRUE, and Netcool Configuration Manager successfully connects to a device using the credentials from the AuthenticationGR, these credentials are cached. The next time Netcool Configuration Manager attempts to connect to that device, it will try to use the cached credentials for that device before looping through the full list in the AuthenticationGR. This will reduce the time taken for Netcool Configuration Manager to connect to devices.

An Authentication GR allows users to specify user login credentials. To edit the default Authentication GR, you must locate it in the Resource Browser, then perform a right-click operation, and select **edit**.

Worker Server GRs

Worker Server GRs contain a series of rules that control which work is permitted to run on which servers.

Worker Server GRs are defined in these terms:

Position
Has no effect in a worker server GR (Position is used by Authentication GRs only)

Server ID
Indicates which server this rule applies to.

The Server ID can be retrieved from Systems Manager/Servers panel by stripping :WORKER suffix

Delay Seconds
Time in seconds that a server will wait before picking up available work

Exclude
Prevents this server from executing work

Ignore
Ignore this rule (used to 'comment out' rules)

Worker Server Algorithm

Each Netcool Configuration Manager worker server builds a cache of what worker servers can execute against each realm. This is calculated every 30 seconds by default, but can be changed in rseries.properties (Core/workerServerUpdatePeriod=30). It is not recommended setting the

workerServerUpdatePeriod to less than 30 seconds. If a new worker server is added to a realm it may be up to 30 seconds before it becomes active.

When the Netcool Configuration Manager worker server grabs a task it firstly checks if it exists in the worker server cache for the realm that the device belongs to. If it does, it then checks which of these worker server GR's VTMOS is a match for the device. The process does not attempt to identify the best VTMOS match; every GR is either a match or is not, and all that are a match are equally valid. The process then inspects the worker server rules defined in the matching GRs that relate to the executing worker server based on the Server ID.

- If the executing worker server has been 'excluded', then the task will not be executed by this worker.
- If the executing worker server has been included, then once the appropriate delay (which could also be zero) has elapsed, the work is executed.

Note: If there are more than one worker server in the deployment and more than one worker is applicable to pick up the work then its the first worker to pick up the work that executes it even if the worker is in a parent realm in the hierarchy. There is no preference given to worker server in the same realm as the device.

Source-based routing

The ITNCM - Base Source Address Definition feature enables an ITNCM - Base defined IP address to be forced as the source address on an IP packet. This controls packet creation, and allows the user to dictate how each packet is handled by way of the source address.

The ITNCM - Base Source Address Definition feature has been implemented to enable source based routing, external to ITNCM - Base, for overlapping IP address spaces. By allowing ITNCM - Base to specify the source IP address, a downstream router can route the packet based on that IP address. This enables network resources in ITNCM - Base to have the same IP address. By segregating these resources by the source IP address, multiple systems with identical IP addresses can be managed by the ITNCM - Base software.

Setting up the Resource Access Doc

For source based routing to work, the IP address must be put into the Resource Access Document (RAD) for the device on which it has been set up. The source address XML tag (`<sourceAddress>`) within the RAD must be specified with the correct IP address.

You must have previously followed the procedure to enable all network routing to go through the default route.

The following procedure explains how to edit the `<sourceAddress>` XML tag to specify the correct IP address that will enable all network routing to go through the default route. You also need to ensure that the `<lbl-mode-flag-forncs>` XML tag is set to the value `true` in order for the RAD to work properly.

To edit the RAD, follow these steps.

1. Go the directory where the RAD is located.
2. Using a text or XML editor, open the RAD for the specified device. The following example opens a RAD called `device1.xml`that has implemented source based routing. This device has an access type of `ssh`.
   - ```
     vi device1.xml
     ```
3. Search for the `<sourceAddress>` XML tag and specify the IP address used to enable all network routing to go through the default route. The following example shows an IP address of `111.111.111.111`.
   - ```
     <sourceAddress>111.111.111.111</sourceAddress>
     ```
4. Search for the `<lbl-mode-flag-forncs>` XML tag and ensure that its value is set to `true`, as in the following example:

- `<lbl-mode-flag-forncs>true</lbl-mode-flagforncs>`

The following example shows the `<sourceAddress>` and `<lbl-mode-flag-forncs>` XML tags in the context of the other XML tags that can reside in a RAD.

```
<access-types>

    <access-type name="ssh">
    <script-id>ssh</script-id>
    <ssh-type>ssh1</ssh-type>

<sourceAddress>111.111.111.111</sourceAddress>
    <streaming-put-flag>false</streaming-put-flag>
    <streaming-get-flag>true</streaming-get-flag>

    <native-compare-flag>false</native-compareflag>

    <import-prewrite-flag>true</import-prewriteflag>

    <sync-prewrite-flag>true</sync-prewrite-flag>

    <import-report-diffs-flag>false</import-reportdiffs-flag>

    <configDataType>CLI</configDataType>

    <reboot-on-config-load>true</reboot-on-configload>

    <import-char-streaming-flag>false</import-charstreaming-flag>
    <import-char-streaming-time-interval>0</import-charstreaming-time-interval>

    <lbl-mode-flag-forncs>true</lbl-mode-flagforncs>
    <user />
    <password />
    <enable-password />
    <prev-user />
    <alt-user />
    <prev-password />
    <alt-password />
    <prev-enable-password />
    <alt-enable-password />
    <hostname />
    <port>22</port>
    <ssh1>
        <cipher>des</cipher>
    </ssh1>
    <ssh2>
        <cipher>blowfish</cipher>
    </ssh2>
</access-type>
```

Resource Access Doc

The Resource Access Doc (RAD) sets up the communication information between the worker server and the actual device. The RAD is one of the most important data structures needed for device interaction. The RAD controls the device connection for IDT and UOWs. ITNCM - Base locates just one RAD when setting up the communication.

The RAD can control every aspect of device interaction, including the connection protocol, timeouts, various workflow elements and scripting.

RADs are created to manage devices at various VTMOS levels. Typically, RADs are built to handle all the devices in an implementation and they will be placed in the base realm for use. All devices that match the VTMOS criteria of the RAD will use it.

ITNCM - Base performs the following tasks when setting up communication for a device:

1. Checks first to see if a custom RAD has been set on the device. A user views a custom RAD by right clicking on a device, and then proceeding to the Properties | Access Tab.
2. If no custom RAD is located, ITNCM - Base checks the realm that the device is in currently to see if a RAD is available for that device.

3. ITNCM - Base then checks all realms, moving upwards in the hierarchical structure until it reaches the top realm.
4. If there is still no match, ITNCM - Base accesses the vendor tree from the database and gets the default RAD for this vendor type.

ITNCM - Base will either match the VTMOS information of the device with the one specified when creating a new RAD, or it will just match Vendor and Type when using the default.

Note: If a RAD GR is created, and custom changes are made, the RAD GUI on the actual device will reflect the changes made.

Every device has a RAD, even it is uses the default RAD that is part of its driver.

A RAD consists of four sections: access order, rollback, access types, and scripts.

1. The access-order describes the order in which the drivers will process the access-types.
2. The access-types set up the communications. The access-type describes the protocol used, and the flag settings. It also contains the script id which indicates if the default or a custom script is being used.
3. The rollback options indicates the type of roll back supported for the device type. This refers to the ability to restore a devices configuration, if an error was received when changing the configuration.
4. The scripts can be used to customise a device script.

Related information

Resource access documents (RADs)
The resource access documents (RADs) sets up the communication information between the Worker server and actual devices on the network. They define the protocol and all the connections to be used.

Editing the RAD

Edit the RAD by using the GUI or XML.

In most cases, it is preferable to create a standalone RAD per VTM (Vendor, Type, Model) rather than a device level RAD due to the maintenance cost. This is because if the RAD were to be edited, every single device would have to be amended instead of just one realm.

Editing the resource access doc with the GUI

Use the GUI to edit the resource access doc (RAD).

Consider creating a stand-alone RAD for each VTM (Vendor, Type, Model) rather than a device level RAD due to the maintenance cost.

Creating a RAD allows you to create an XML document which is used to communicate with devices on the network. The following steps describe how to display the Resource Access Data dialog.

1. Click **File** > **New** and select **Resource Access**.
2. The **New Network Resource** dialog box displays.
3. Specify values for the **New Network Resource** dialog box using the following table as a guide:

Element	Description
Name	Type a name for the resource that you are creating.
Vendor	Type the vendor name for the resource you are creating.
Type	Type a type name for the resource you are creating.

Element	Description
Model	Select an available model from the drop down list for the resource you are creating. Some models allow you to use wild cards.
OS	Select an available operating system from the drop down list for the resource you are creating. Some operating systems can be specified by using a wildcard.

4. Click **OK**. The RAD specified is created.
5. To edit the RAD for a specific device, select **Resource Browser** from the navigation tree. The resources are displayed in the pane on the right hand side.
6. Select or Search for a device. See the ITNCM User Guide for information on searching within ITNCM - Base.
7. Right click the Device, and Select Resource Access. The Resource Access Doc is displayed.
8. Click Edit to make changes to the document. The Resource Access Data dialog is displayed.

From the Resource Access Data dialog, you can perform these tasks:

- Set up access types
- Set up transport information
- Set up SSH options
- Set up authentication
- Set up command line information
- Set up configuration information
- Set up scripts
- Set up rollback options

Setting up access types

Set up access types by selecting the **Access Types** tab from the Resource Access Data dialog.

You must have followed the steps to display the Resource Access dialog.

The access type determines how to communicate with a device. From the **Access Types** tab, a list of access types for the chosen device is presented in the pane on the left hand side. The following steps describe how to set up access types.

1. Select the **Access Types** tab from the Resource Access Data dialog.
2. Select an access type from the pane on the left hand side. The name of the access type selected is populated in the Type Name field.
3. Select the Enabled check box to enable the selected access type.
4. There are a number of configurable aspects to the creation of a RAD, which must be set up. These are shown as sub-tabs displayed in a vertical fashion.

There are a number of set up tasks associated with the creation and editing of a RAD. The set up tasks are shown as sub-tabs displayed in a vertical fashion. Make the appropriate selection according to the following table:

Set up task	Selection tab
Set up transport information	Select the **Transport** sub-tab.
Set up SSH options	Select the **SSH Options** sub-tab.
Set up authentication	Select the **Authentication** sub-tab.

Set up task	Selection tab
Set up command line information	Select the **Command Line** sub-tab.
Set up configuration information	Select the **Configuration** sub-tab.

Setting up transport information
The transport page deals specifically with the details used to make a connection to the device. It also defines methods by which data will be retrieved and placed on the device. Set up transport information by selecting the **Transport** sub-tab from the Resource Access Data dialog.

You must have followed the steps to display the Resource Access dialog and then to select an access type from the **Access Types** tab.

The following steps describe how to set up transport information.

1. Select the **Transport** sub-tab from the Resource Access Data dialog.
2. The transport-related fields display on the dialog. Use the information in the following table to specify appropriate values for these fields.

Option	Description
Transport field	Description
Hostname	This hostname is used to define a terminal server by which to connect to a device through a console port. Type the hostname for the device being accessed.
Port	This port number typically represents the ports on the device that are listening for either ssh or telnet connections, for example, 23 for telnet and 22 for ssh. Type the port for the device being accessed.
Source Address	The source address is used when source routing is used in the design. Type the source address for the device.
Connection Timeout	The connection timeout is used to determine how long to wait for a TCP connection to a socket on the network device, This timeout is used when sending the connect prompt to a device. This is used to see if you still have an active connection.
Streaming	Streaming determines how data is placed on the device. By default it is unchecked, which means that a file transfer protocol will be used. Select the type of streaming you require.
Character	This is used for sending a character at a time.
Timeout	If you select the 'Character' checkbox then this will apply. Some devices require ITNCM - Base to send one character at a time when logging in.
Connect Response Timeout	This is the timeout for the connection to be established and the user to successfully login.
Connect Retry Delay	This is the time between each connection retry, only if the option to use a retry has been selected.

Option	Description
Max Response Timeout	This is maximum time that a response will be waited for from a device.
Response Timeout	This is the time waited for a single wait statement. This is continually reset when reading the buffer and receiving the token. The maximum response timeout will override this if it is affected while retrieving data. Some devices keep sending data and we keep resetting the response time out, so the maximum response time out is used to override this and prevent an infinite loop.

3. Click the Save icon, to save any changes made.

Make the appropriate selection according to the following table:

Set up task	Selection tab
Set up SSH options	Select the **SSH Options** tab.
Set up authentication	Select the **Authentication** tab.
Set up command line information	Select the **Command Line** tab.
Set up configuration information	Select the **Configuration** tab.

Setting up SSH options
The SSH options page allows the RAD to determine the specifics of the SSH protocol that should be used to connect to the devices. Set up SSH options by selecting the **SSH Options** sub-tab from the Resource Access Data dialog.

You must have followed the steps to display the Resource Access dialog and then to select an access type from the **Access Types** tab.

The following steps describe how to set up SSH options.

1. Select the **SSH Options** sub-tab from the Resource Access Data dialog.
2. The SSH-related fields display on the dialog. Use the information in the following table to specify appropriate values for these fields.

Option	Description
SSH Option	Description
SSH Type	Select from the available SSH types.
SSH1 Preferred Cipher	This is the preferred cipher or comma delimited list of ciphers that will be used for SSH1. Autonegotiation will occur if the preferred cipher is not available or this field is blank.
SSH2 Preferred Cipher	This is the preferred cipher or comma delimited list of ciphers that will be used for SSH2. Autonegotiation will occur if the preferred cipher is not available or this field is blank.

3. Click the Save icon, to save any changes made.
4. The following cipher names will be transformed into complete ciphers names. All other cipher names will remain unaltered.

Specified Partial Cipher Name	Actual Cipher Name Used
aes128	aes128-ctr

Specified Partial Cipher Name	Actual Cipher Name Used
aes256	aes256-ctr
des or des3	3des-ctr
blowfish	blowfish-ctr
idea	idea-cbc

Make the appropriate selection according to the following table:

Set up task	Selection tab
Set up authentication	Select the **Authentication** tab.
Set up command line information	Select the **Command Line** tab.
Set up configuration information	Select the **Configuration** tab.

Setting up authentication
Set up authentication by selecting the **Authentication** sub-tab from the Resource Access Data dialog. It is possible to store device authentication credentials in the RAD, but this is not considered best practice. The Auth GR is recommended for storing credentials.

You must have followed the steps to display the Resource Access dialog and then to select and access type from the **Access Types** tab.

The following steps describe how to set up authentication.

1. Select the **Authentication** sub-tab from the Resource Access Data dialog.
2. The authentication-related fields display on the dialog. Use the information in the following table to specify appropriate values for these fields.

Option	Description
Authentication field	Description
Current	This provides the authentication information for the current config.
Username	Type the required username to access the device.
Password	Type the required password to access the device.
Enable Password	Type the enable password to access the device.
Previous	This provides the authentication information for the previous config.
Username	Type the required username to access the device.
Password	Type the required password to access the device.
Enable Password	Type the enable password to access the device.
Alternate	This provides alternate authentication information.
Username	Type the required username to access the device.

Option	Description
Password	Type the required password to access the device.
Enable Password	Type the enable password to access the device.

3. Click the Save icon, to save any changes made.

Make the appropriate selection according to the following table:

Set up task	Selection tab
Set up command line information	Select the **Command Line** tab.
Set up configuration information	Select the **Configuration** tab.

Setting up command line information
The command line information page defines which script will be used by the particular access type chosen. Set up command line information by selecting the **Command Line** sub-tab from the Resource Access Data dialog.

You must have followed the steps to display the Resource Access dialog and then to select an access type from the **Access Types** tab.

The following steps describe how to set up command line information.

1. Select the **Command Line** sub-tab from the Resource Access Data dialog.
2. The command line-related fields display on the dialog. Use the information in the following table to specify appropriate values for these fields.

Option	Description
Command line field	Description
Script	This value is usually set to either **`default`** (when telnet access types are enabled), or **`ssh`** (when ssh access types are enabled). A user defined script name can be used. However, that name must be either defined in the RAD scripts or defined in the Script resource. Select your required script from the available drop-down.
Prompt	This is used for socket checking when working in the device's initial login mode. This will override the defaults that are contained in the script set that is resolved for this device. Type in the required prompt.
Enable Prompt	This is used for socket checking when working in the device's privileged mode. Type in the required enable prompt.
Configuration Edit Prompt	This is used for socket checking when working in the device's configuration mode. Type in the required edit prompt.

3. Click the Save icon, to save any changes made.

Make the appropriate selection according to the following table:

Set up task	Selection tab
Set up configuration information	Select the **Configuration** sub-tab.

Setting up configuration information
The configuration page controls how UOWs are executed for devices that resolve this RAD. These configurations only affect the access type you are working on. Set up configuration information by selecting the **Configuration** sub-tab from the Resource Access Data dialog.

You must have followed the steps to display the Resource Access dialog and then to select an access type from the **Access Types** tab.

The following steps describe how to set up configuration information.

1. Select the **Configuration** sub-tab from the Resource Access Data dialog.
2. The configuration-related fields display on the dialog. Use the information in the following table to specify appropriate values for these fields.

Option	Description
Native Compare	This changes the algorithm Netcool Configuration Manager uses to calculate differences between two device configurations. The algorithm is used both when processing UOWs and when displaying differences to end-users. By default this option is not enabled, causing Netcool Configuration Manager to calculate differences using an internal tree produced as part of the SmartModel. Enabling this option will cause Netcool Configuration Manager to perform a raw textural difference instead, and this can produce false positives because a device configuration can change texturally without changing semantically. For example, some Cisco devices have an embedded pseudo-configuration setting that is related to clock time. When a device is imported, if the RAD uses native compare, the difference is calculated in native format. If a change is detected, then changes are analysed again using the SmartModel, and the modeled version of the change is displayed to the user. When a device is updated, the diffrence is calculated in native format. If no change is detected, the update is applied. If a change is detected, the update is not applied, and changes are analysed again using the SmartModel, and the modeled version of the change is displayed to the user. There are two situations that may require you to enable this option: • The RAD is being used by a Standard Driver or a Custom Driver. This situation always requires you to ensure the Native Compare option is enabled. • The device configuration is extremely large to the point where the tree-based algorithm is infeasible.
Reboot on Config Load	This enables the device to reboot after loading the new config. This determines the behaviour when disaster recovery UOW, or certain rollback options are selected.
Allow Line by Line for Native Command Set	When enabled, this flag allows line by line mode to be enabled when applying a native command set. Line by line mode checks the buffer for errors between each command submission and will disable FTP processes.
Compare Device and ITNCM	This compares the device's running configuration with what is stored in the Netcool Configuration Manager database. If differences are found, they are reported and the change to the device does not take place.

Option	Description
Multiple Config Compare	This causes the running and stored configurations to be retrieved and compared. If differences are found, they are reported and the change to the device does not take place.
Disable Config Checks on Changes	This in conjunction with System Properties options causes Netcool Configuration Manager to bypass any checks for differences in the running and stored configurations as well as the device running config and the Netcool Configuration Manager stored config.
Prewrite	This causes a copy of the running config to be copied to the startup config prior to an import sync (or both, based on your selection).
Report Diffs	This enables reporting of differences between the stored and running configurations. The prewrite option, if enabled, is performed after the retrieval of the configurations.
Update on Change	This enables the update on changes of info, configuration or both.
Config Data Type	This lists the available config data types. For example, CLI, BIN, CLIandBIN.

3. Click the **Save** icon, to save any changes made.

Make the appropriate selection according to the following table:

Set up task	Selection tab
Set up scripts	Select the **Scripts** tab.

Setting up scripts

Set up scripts by selecting the **Scripts** tab from the Resource Access Data dialog.

You must have followed the steps to display the Resource Access dialog.

The following steps describe how to set up scripts.

1. Select the **Scripts** tab from the Resource Access Data dialog. The script editor is displayed.
2. Make changes as required.

Make the appropriate selection according to the following table:

Set up task	Selection tab
Set up rollback options	Select the **Rollback Options** tab.

Setting up rollback options

Set up rollback options by selecting the **Rollback Options** tab from the Resource Access Data dialog.

You must have followed the steps to display the Resource Access dialog.

The following steps describe how to set up rollback options.

1. Select the **Rollback Options** tab from the Resource Access Data dialog.
2. The Rollback screen is displayed. Use the information in the following table to specify appropriate values for these fields.

Option	Description
Rollback option	Description

Option	Description
NO_ROLLBACK	Leave as is.
MODELLED_ROLLBACK	Use XML models to determine how to restore the device.
SPECIAL_ROLLBACK	This is used for Juniper devices that run with XML.
REBOOT_ROLLBACK	This pushes the original config to the startup config on the device prior to a device restart.

3. Select either the Recommended or Required check box as a rollback status.
4. Click the Save icon once changes have been made.

Device scripts

ITNCM - Base uses device scripts to determine the work on which to proceed.

A device script can be used to customize scripts to address issues with a particular network device.

A device script consists of the following items:

- Device script sections — The required section of the device script performs the functions needed. For example, the default.error section lists string errors. Each section of the device script specifies the commands in the order that they will be executed. If a special command (if then else) is in the section at the correct time, this will be constructed and the data will be checked against them.
- Device script variables — Each device script makes use of variables, for example, `$connect_username$`, `$alt_username$`, `$connect_password$`, and so forth.
- Device script commands — Each device script contains a list of the show commands that will be sent to the device. The various show commands are used to determine the hardware makeup of the device.

Note: A device script is in a loop.

The `<script-id>` element, specified in the RAD, identifies the device script that ITNCM - Base uses. Multiple device scripts (each with their own name) may be included within the RAD. However, only the one being referenced will be used. By convention, specifying the name `default` within the `<script-id>` element means the default device script stored in the database for that device (based on VTMOS) will be used.

The following briefly describes how ITNCM - Base interacts with a device script:

- The `commHandler` sends a command and then waits to determine if something comes back. As the information is coming back it is received in packets and placed in a buffer. (The packet may or may not contain all the data, so many packets could come in.)
- Each time a packet is received, ITNCM - Base reads the data and checks to see if there are more packets. ITNCM - Base will do this a number of times, and then stop to check for errors in the buffer.
- If no errors are found, ITNCM - Base will look for the required token. If the token is not found, then ITNCM - Base checks to see if further processing is required (the if then else).
- If this processing is required, ITNCM - Base will carry this out before it returns to reading the packets. ITNCM - Base will continue to read the packets until the token is found, an error occurs, or a predefined timer is hit.
- Once a token is found then the special commands are taken down. It is important to realize that ITNCM - Base can go through a section many times until the token is found. This can cause ITNCM - Base to execute the same if then else command many times depending on the script.

Device script sections

Each device script consists of sections that perform specific tasks as required.

Section names and descriptions

The following table provides the names and descriptions of the device script sections.

Section name	Description
name of script	Lists the name of the script. The default script uses: • ssh • ssh-end • name=
default.prompt	The prompt in device enable mode used to show a good connect.
default.error	Lists string errors.
connect	Describes the steps to setup and make a connection.
diag	Provides the ability to list any device information required, such as hardware type, and so forth.
model	Gives the model information.
copyRunning	Copies configs to startup config and ensures synchronization.
fileRunning	Informs the device to transfer it's config to an FTP server.
fileStored	Gets stored config off device in file format.
config.running	Gets running config via streaming.
config.stored	Gets stored config via streaming.
config.version	Gets the version of the config
config.start	Gets the start of the config.
config.end	Gets the end of the config.
disconnect	Describes the steps required to disconnect from a device.
ftp	Used for copying new configs or snippets to a device.
streamFile	Enables down streaming.
fileFtpToStartUp	Pushes config to startup (used for disaster recovery).
fileStreamToStartUp	Streams config to startup.
notify	Netcool Configuration Manager sends device terminal broadcast messages, such as "Pulling configuration to store in ITNCM", which are configured in the RAD scripts that are used to communicate with devices.

Section name	Description
	To stop these messages, comment out the 'notify' section of the appropriate RAD using the hash (#) character.
dir	Lists directories.
del	Used for deleting a file.
cd	Used for changing a directory.
squeeze	Used for squeezing a file system.
copyDown	Enables copying down to the system.
erase	Erases a file system.
download	Enables downloading using a router's default capabilities.
reload	Enables reloading to the box.

Device script variables

Each device script makes use of variables.

Global variables

The following global variables can be placed at the start of a Device Script:

REMOVE-LAST-LINE-CONFIG

This will remove the last line of a configuration as deliminated by config.running.FIND-BEGIN and config.running.FIND-END. This is useful in the case where there is no easy way to avoid the capture of the last line.

Example:

```
REMOVE-LAST-LINE-CONFIG=true
```

replaceAll

These global variables are used to replace parts of each line returned by the device before Netcool Configuration Manager performs any further processing or storing of the configuration. The separator between the regular expression and the new value (/ below) is taken to be the first character after the =.

```
replaceAllVersionInfo=/regex pattern/new value/
replaceAllConfigInfo=/regex pattern/new value/
replaceAllDiagInfo=/regex pattern/new value/
replaceAllModelInfo=/regex pattern/new value/
replaceAllSendCmdInfo=/regex pattern/new value/
```

replaceAllDiagInfo operates on the device response whilst executing the diag subscript.

replaceAllVersionInfo and replaceAllModelInfo operate on the response when executing the configVersion and model subscripts respectively,

replaceAllConfigInfo operates on any subscript that fetches the device configuration.

replaceAllSendCmdInfo operates on any device responses to Native Command Sets.

Multiple replaceAll lines can be specified by incrementing the counter for each of them. For example to eliminate all "--More--" prompts and all cursor home terminal escape sequences:

```
replaceAllDiagInfo . 00 =/\-\-More\-\-//
replaceAllDiagInfo . 01 = / \u001B\\[24;\[\\d\]*H //
```

The following example removes any line containing 'password' when Netcool Configuration Manager fetches the device configuration (for example config.runnning):

```
replaceAllConfigInfo=/.*password.*\r\n//
```

replaceChars

These global variables perform the same as replaceAll variables. but are exact character matches instead of using regular expressions. The following are being phased out. Below are examples of the commands that are used in the device scripts.

```
replaceCharsDiagInfo.00=/\u0016/ /
replaceCharsConfigInfo.00=/\u0016/ /
replaceCharsModelInfo.00=/\u0016/ /
replaceCharsVersionInfo.00=/\u0016/ /
replaceCharsSendCmdInfo.00=/\u0016/ /
```

sshAuthType

The variable lets the ssh driver know what type of authentication request is used. By default it uses password and will try both if needed. The values can be: {password, KbdInteract}

turnOffAutoRetry

The driver tries to connect to the device twice if the 1st username set is incorrect. This happens for every incorrect username/password set. Netcool Configuration Manager tries the connection the second time for robustness. When the driver gets an IO error during the first connection, then the driver will retry the connection a second time. This happens if the device is busy or there is some kind of other network glitch. The auto retry can be turned off by setting turnOffAutoRetry=yes in the device script. The value can be: {yes, no}

numCiphersToTry

Use this variable to set the number of additional ciphers to try after using the cipher or ciphers specified in the Resource Access Document. Users can specify a comma delimited set of ciphers and the application will try each to connect to the device.

For example, `aes256-cbc`, `aes128-ctr`.

If only the specified ciphers are to be used, set the `numCiphersToTry=0`. The default is three.

sshTermType

This sets the term type that the ssh driver uses. The term type can be dumb (default same as it has always been), xterm, linux, scoansi, att6386,sun, aixterm, vt220, vt100, ansi, vt52, xtermcolor, linux-lat, at386, vt320, vt102, Tandem 6530

The following device script variables are applicable only to file-based operations.

getConfigFileServer

This tells Netcool Configuration Manager that the configuration will be retrieved from an ftp/scp server as a file.

getModelFileServer

This tells Netcool Configuration Manager that the model information will be retrieved from an ftp/scp server as a file.

getDiagFileServer

This tells Netcool Configuration Manager that the diag information will be retrieved from an ftp/scp server as a file.

getVersionFileServer

This tells Netcool Configuration Manager that the version information will be retrieved from an ftp/scp server as a file.

getBinaryFileServer

This tells Netcool Configuration Manager that the binary configuration information will be retrieved from an ftp/scp server as a file.

InfoScpFile

This tells the driver to get the latest or oldest file that is found to match the command that is supplied in the getFtpFileName command. It can be one of the following {LATEST, OLDEST }

The default value is latest.

Variables

The following table provides the names of the variables that device scripts can use.

Variables	Description

$connect_username$	This variable is used during connection and is substituted with the value from the default database, authentication resource, RAD, or the submitted UOW if set. The following example is from a device script:
	`connect.05.then=send=`
	`$connect_username$\r`

$alt_username$	This variable is used during connection and is substituted with the value from the default database, authentication resource or RAD if set.
$connect_password$	This variable is used during connection and is substituted with the value from the default database, authentication resource, or the RAD if set.
$alt_password$	This variable is used during connection and is substituted with the value from the default database, authentication resource, or the RAD if set.

$ftp_filename$	This variable is used when getting/putting device info by ftp/cps, and is substituted with the random file name the driver creates, or a value that is set by the device script commands.
	For
	example:
	`fileRunning.01.send=copy running-config`
	`ftp://$ftp_username$:$ftp_password$@$`
	`ftp_hostname$/$ftp_filename$\r`

$ftp_hostname$	This variable is used when getting/putting device info by ftp/scp and is substituted with the value from the default database or the file transfer resource. example from device script.
	`fileRunning.01.send=copy running-config`
	`ftp://$ftp_username$:$ftp_password$@$`
	`ftp_hostname$/$ftp_filename$\r`

$ftp_username$	This variable is used when getting/putting device info by ftp/scp and is substituted with the value from the default database or the file transfer resource. example from device script.
	`fileRunning.01.send=copy running-config`
	`ftp://$ftp_username$:$ftp_password$@$`
	`ftp_hostname$/$ftp_filename$\r`

$ftp_password$	This variable is used when getting/putting device info by ftp/scp and is substituted with the value from the default database or the file transfer resource. example from device script.

Variables	Description
	`fileRunning.01.send=copy running-config ftp://$ftp_username$:$ftp_password$@$ ftp_hostname$/$ftp_filename$\r`
$ftp_althostname$	This variable is used when getting device info by ftp/scp and is substituted with the value from the default database or the file transfer resource. example from device script. `copyDown.01.send=copy ftp://$ftp_altusername$:$ftp_altpassword $@$ftp_althostname$/$ftp_altpath$/ $copy_input1$ $copy_input2$\r`
$ftp_altusername$	This variable is used when getting device info by ftp/scp and is substituted with the value from the default database or the file transfer resource. example from device script. `copyDown.01.send=copy ftp://$ftp_altusername$:$ftp_altpassword $@$ftp_althostname$/$ftp_altpath$/ $copy_input1$ $copy_input2$\r`
$ftp_altpassword$	This variable is used when getting device info by ftp/scp and is substituted with the value from the default database or the file transfer resource. example from device script. `copyDown.01.send=copy ftp://$ftp_altusername$:$ftp_altpassword $@$ftp_althostname$/$ftp_altpath$/ $copy_input1$ $copy_input2$\r`
$message$	This variable is substituted by a message from the driver when sending the message that Netcool Configuration Manager is doing some work on the device. `notify.message.send=$message$`
$enable_password$	This variable is used during connection and is substituted with the value from the default database, authentication resource, RAD, or the submitted UOW if set.
$alt_enable_password$	This variable is used during connection and is substituted with the value from the default database, authentication resource, or the RAD if set.
$action$	This substitutes the action that will be done (replace/merge) on the device. Used for juniper devices. `ftp.11.send=<rpc><load-configuration action="$action$"url="/tmp/$ftp_filename $"/></rpc>\r`
$stream_input$	This variable is used when streaming data to the device. It usually has the config or config snippet that is getting changed. `streamFile.03.sendData=$stream_input$\r`
$copy_input1$	This variable is used when copying data to the device. Used most of the time in os upgrade. The value comes from the os spec.

Variables	Description

	`copyDown.01.send=copy ftp://$ftp_altusername$:$ftp_altpassword $@$ftp_althostname`
	
	`$/$ftp_altpath$/$copy_input1$ $copy_input2$\r`

$copy_input2$	This variable is used when copying data to the device. Used most of the time in os upgrade. The value comes from the os spec.
	
	`copyDown.01.send=copy ftp://$ftp_altusername$:$ftp_altpassword $@$ftp_althostname`
	
	`$/$ftp_altpath$/$copy_input1$ $copy_input2$\r`

del_input	This variable is the file that will be deleted from a device. The value comes form the OS spec, and is used when making needed room on the device for the new OS image. `del.01.send=del $del_input$\r`
dir_input	This variable is used when getting file info from the device. This is used for OS upgrades. `dir.01.send=dir $dir_input$:\r`
cd_input	This variable is used to change directories on the device, usually used when doing an os upgrade and need to put a file on the device. `cd.01.send=cd $cd_input$:\r`
$squeeze_input$	This is used when doing an upgrade on some device. `squeeze.01.send=squeeze $squeeze_input$:\r`
$erase_input$	This is used when doing an upgrade and ITNCM has been instructed to erase the file system on the device. `erase.01.send=erase $erase_input$:\r`
$down_load_input$	This variable is used when doing an OS upgrade. This value comes for the OS spec. `download.01.send=copy ftp://$ftp_username$:$ftp_password$@$ ftp_hostname$/$down_load_input1$ $down_load_input2$\r`
$prompt$	Used when getting a dynamic prompt. `model.00.getPrompt1=\r model.01.send=show switch\r model.end=$prompt1$`
$enablePrompt$	This variable is used in cases where the enable prompt is changed. Not currently used.

Variables	Description
$configEditPrompt$	This variable is used in cases where the config edit prompt is changed. Not currently used.
$binaryDataFileName$	This variable is used when setting the binary data file to a given name. Not currently used.

$address$	This variable is the device address after a lookup. Used when getting and putting config info and reconnecting.
	`getBinaryData.05.exec=sh`
	`<<insert path and script name here>> $ftp_filename$`
	`$address$ $connect_username$`
	`$connect_password$ flash/snapshot-config`
	`$ftpPath$\r`

$addressName$	This variable is the reverse lookup name of the address. It is used when the name is needed.
	`config.running.01.getFtpFileName=ls -r`
	`$ftpPath$/*$addressName$* |`
	`cut -d"/" -f4,first`

$fileName$	This is the file name that the driver created or was set in the device script to contain info that is being sent or received by the driver.
	`diag.02.setRandomFileName=true`
	`diag.03.writeReturnBufftoFile=`
	`$ftpPath$\\$fileName$`

$ftpFileName$	This is the file name that the driver created or was set in the device script to contain info that is being sent or received when doing ftp/scp by the driver. Not currently in a device script.

$ftpPath$	This is the variable is substituted with info from the file transfer object ftp path info.
	`fileRunning.01.send=copy running-config`
	`ftp://$ftp_username$:$ftp_password$@$`
	`ftp_hostname$/$ftpPath$/$ftp_filename$\r`

$ftp_altfqpath$	This variable is substituted with info from the file transfer object altFtpInfo section and is used when doing OS upgrades.
	`scpcopyDown.01.send=copy file`
	`scp://$ftp_altusername$@`
	`$ftp_althostname$`
	`/$ftp_altfqpath$/$copy_input1$`
	`$copy_input1$\r`

$ftp_altpath$	This variable is substituted with info from the file transfer object altFtpInfo section and is used when doing OS upgrades.
	`scpcopyDown.01.send=copy`
	`scp://$ftp_altusername$:$ftp_altpassword`
	`$@$ftp_althostname$/$ftp_altpath$/`
	`$copy_input1$`
	`$copy_input2$\r`

$ctrlA$	
$ctrlB$	

Variables	Description
$ctrlC$	
$ctrlD$	
$ctrlE$	
$ctrlF$	
$ctrlN$	
$ctrlO$	
$ctrlP$	
$ctrlU$	
$ctrlV$	
$ctrlW$	
$ctrlX$	
$ctrlY$	
$ctrl[$	
$ctrlSLASH$	
$ctrl-$	
$ctrl@$	
$ctrlZ$	
$ctrl]$	
$ctrl^$	

Device script commands

Each device script makes use of commands.

Commands

The following table provides the names of the commands that device scripts can use.

Table 1. Device script commands

Commands	Description

clearReturnBuff=<true,false>	This command clears the return buff of all data. This is used when data is already present in the buffer and it is not needed.
	`connect.15.wait=name:connect.17.`
	`clearReturnBuff=true`
	`connect.20.send=$connect_username$\r`

exec=cmd	This command allows the driver to process external commands. In the following example a script in the drivers bin directory is executed, and the results are put into the return buffer.
	`putBinaryData.01.exec=sh`
	`/opt/IBM/tivoli/netcool/ncm/drivers/bin/`
	`exampleScript.sh`
	`$ftp_filename$ $address$`
	`$connect_username$ $connect_password$`
	`flash/snapshot-config`
	`$ftpPath$\r`

getFtpFileName=<cmd>,<first,last>	This command gets a list of files from the command that is executed on the ftp/scp server via sshexec. The last field lets the driver get the last or 1st of the files.
	`config.running.01.getFtpFileName=ls -r`
	`$ftpPath$/*$addressName$* | cut -d"/"`
	`-f4,first`

getSCP=<username>:<password>@
<host>:
<path><filename>
	This command gets a file from an ftp/scp server and imports the info as if the driver connected to a device and obtained the info.
	`config.running.02.getSCP=$ftp_username$:`
	`$ftp_password$@$ftp_hostname$:`
	`$ftpPath$/$ftp_filename$`

if-then-else	This command uses standard if-then-else logic to run nested commands based on logical tests.
	In the following example, the script looks for a prompt from the device that ends in ame. If this string is found, then the script responds with the username. If the string as is found instead, then the script responds with the username and password.
	`connect.01.if=ame:`
	`connect.05.then=send=$connect_username$\r`
	`connect.06.then=sleep=1000`
	`connect.10.elseIf=as:`
	`connect.15.then=send=$connect_username$\r`
	`connect.16.then=sleep=100`
	`connect.25.then=wait=word:`
	`connect.30.then=send=$connect_password$\r`
	`connect.35.then=sleep=1000`
	`connect.36.endIf`

Table 1. Device script commands (continued)

Commands	Description
ignoreErrors=true/false	This lets the driver ignore an error that is thrown and will continue processing. `model.01.ignoreErrors=true`

maxResponseTimeout=time millsec	This command lets the device script override the maxResponseTimeout that is set in the system, or RAD. This is usually used in sections where the time might be longer, but you do not want to change it for every device or commands.
	`copyDown.00.maxResponseTimeout=3000000`
	`copyDown.01.responseTimeout=2000000`
	`copyDown.02.send=copy ftp://`
	`$ftp_altusername$:$ftp_altpassword`
	`$@$ftp_althostname`
	`$/$copy_input1$ $copy_input2$\r`
	`copyDown.03.wait=?`
	`copyDown.04.send=\r`
	`copyDown.05.wait=]`
	`copyDown.06.send=n\r`
	`copyDown.07.wait=#`
	`copyDown.08.maxResponseTimeout=300000`
	`copyDown.09.responseTimeout=600000`

modelMaxSize=integer	This command sets the maximum number of characters for a model. This is used to catch errors with the device script when getting model info. The default is 35. `Model.00.modelMaxSize=60`

polling=count=<number of loops>,timeout=<time waiting for a response>,duration=<sleep time>,token=<some token>	This command polls a device waiting for a token. It allows the driver to sleep while it is waiting.
	`copyDown.15.polling=count=1000,`
	`timeout=2000,duration=2000,token=#`

pollwithreload=count=<number of loops>,duration=<sleep time>	This command polls the device reconnecting after a rest/reboot/reload was executed.
	`reload.06.pollwithreload=count=1000,`
	`duration=1200`

readReturnFromFile=<path><file>	This command reads a file and returns it in the return buff as if the device returned the data. Not used in any of the device scripts at this time.
reConnect=true,false	This command reconnects the driver to the device. `ftp.06.reConnect=true\r`

Table 1. Device script commands (continued)

Commands	Description
reConnectWithAltHost=true,false	This command reconnects the driver to the device with the alt host ID. `Not used now`

responseTimeout=time milsec	This command lets the device script override the responseTimeout that is set in the system, or RAD. This is usually used in sections where the time might be longer. but you do not want to change it for every device or commands.
	`copyDown.00.maxResponseTimeout=3000000`
	`copyDown.01.responseTimeout=2000000`
	`copyDown.02.send=copy ftp://$ftp_altusername$:$ftp_altpassword`
	`$@$ftp_althostname`
	`$/$copy_input1$ $copy_input2$\r`
	`copyDown.03.wait=?`
	`copyDown.04.send=\r`
	`copyDown.05.wait=]`
	`copyDown.06.send=n\r`
	`copyDown.07.wait=#`
	`copyDown.08.maxResponseTimeout=300000`
	`copyDown.09.responseTimeout=600000`

saveReturnBuff=<true,false>	This command saves the return buff from a device to be used later.
	`diag.00.callAPI=SnmpRequest -OtCSV -v 1 $address$ 1.3.6.1.2.1.1.1 \r`
	`diag.01.saveReturnBuff=true`
	`diag.02.setRandomFileName=true`
	`diag.03.writeReturnBufftoFile= $ftpPath$\\$fileName$`

send=some string	This command sends data to the device. It is used throughout the device script. `connect.75.send=$enable_password$\r`
sendData=some string	This is the same as sent with the added inline command processing. To use the inline commands, you change the send command to sendData in the streaming config section. This is used for config change native command sets only. If in interrogation mode, then the driver will communicate directly with the device without going through the sendData command. `streamFile.03.sendData=$stream_input$\r`

Table 1. Device script commands (continued)

Commands	Description
sendSaveReuse=command	This command sends data to the device, but it determines if the data was already received. If the data was received, then the driver uses that data without returning to the device. If the data was not, then the driver will retrieve it from the device. `diag.01.sendSaveReuse=show version\r`
setFileName=filename	This command sets the ftpFileName to a value.

setFtpFileName=filename	This command sets the ftpFileName to a value.
	`#getBinaryData.00.setFtpFileName=`
	`config.cnf`

setFtpPath=path	This command sets the ftpPath variable to be used when getting and putting info with ftp/scp.

setRandomFileName=true,false	This command sets a random name for the $fileName$ variable.
	`config.version.03.setRandomFileName=true`
	`config.version.04.writeReturnBufftoFile=`
	`$ftpPath$\\$fileName$`

setRandomFtpFileName=true,false	This command sets a random name for the ftpFileName variable. Not used in any device script now.

setReturnBuff=some str	This command sets the return buff to info as if the device passed it to the driver. If this is used without one of the setXXXFileServer variables set, then the driver will process it with the 'find begin' and 'find end' variables. The following examples are first with the processing, and then without.
	With processing
	`# Model`
	`model.setReturnBuff=$##11000##`
	`model.send=\r`
	`model.end=#`
	`model.FIND-BEGIN=##`
	`model.FIND-END=##`
	Without processing
	`model.01.setReturnBuff=MRS-BGF`
	`model.end=$`
	`model.FIND-BEGIN=`
	`model.FIND-END=`

Table 1. Device script commands (continued)	
Commands	**Description**
sleep=time milsec	The sleep command lets the drivers sleep for the specified number of milliseconds before resuming processing. It is used throughout the device scripts, and especially in the connect section to let the device get to the next state before results are checked. `connect.00.sleep=1000`
wait=some string	This command waits for data from the device. It is used throughout the device script. `connect.85.wait=#`

writeReturnBufftoFile=<path><file>	This command writes the return buff to a file to be processed externally or used at a later date.
	`diag.00.callAPI=SnmpRequest -OtCSV -v 1 $address$ 1.3.6.1.2.1.1.1 \r`
	`diag.01.saveReturnBuff=true`
	`diag.02.setRandomFileName=true`
	`diag.03.writeReturnBufftoFile= $ftpPath$\\$fileName$`

Dynamic prompts

Two prompt variables can be used to store the device command prompt, and then later use it for purposes such as a deliminator when extracting device responses. They are *$prompt1$* and *$prompt2$*, and are accessed by **`getPromptN`** and **`setPromptN`** commands.

Commands

First you initialize the prompt variable, for example:

```
config.running.00.getPrompt1=\r
```

The argument to `getPrompt1` is a string to send the device that will result in the device returning the content destined for the prompt variable. Once initialized, it may be used elsewhere, as illustrated in this example:

```
config.running.00.getPrompt1=\r
config.running.01.send=show running-config\r
config.running.end=$prompt1$
config.running.FIND-BEGIN=Running
config.running.FIND-END=$prompt1$

config.running.01.send=config t\r
config.running.03.wait=)#
config.running.04.getPrompt1=\r
config.running.05.send=show running-config\r
config.running.08.sleep=6000
config.running.end=$prompt1$
config.running.FIND-BEGIN=configure
config.running.FIND-END=$prompt1$
```

The device scripts in this example determine the device prompt by sending it a carriage return, and then uses it to determine the end of `config.running` execution, as well as the end of the text to extract that represents the device configuration.

postConnect and IDTPostConnect commands

When processing UOWs and connecting users to devices via IDT, Netcool Configuration Manager uses the applicable connect section of the Device Script.

- In all cases, the connect.* commands will be used.
- postConnect commands are not passed to IDT and used only when executing UOWs.
- IDTPostConnect commands are only processed by IDT.

For example, to make IDT ignore the term length and term width commands, change "connect.xx.send=" entries into "postConnect.xx.send="

Original connect section

```
connect.81.wait=#
connect.82.send=term len 0\r
connect.85.wait=#
connect.90.send=term width 100\r
connect.95.wait=#
```

New Connect section

```
connect.81.wait=#
postConnect.82.send=term len 0\r
postConnect.85.wait=#
postConnect.90.send=term width 100\r
postConnect.95.wait=#
```

Furthermore, to make IDT set user-orientated term length and width, add "IDTPostConnect.xx.send=" entries to the connect section.

```
connect.81.wait=#

postConnect.82.send=term len 0\r
postConnect.85.wait=#
postConnect.90.send=term width 100\r
postConnect.95.wait=#

IDTPostConnect.96.send=term len 24\r
IDTPostConnect.97.wait=#
IDTPostConnect.98.send=term width 80\r
IDTPostConnect.99.wait=#
```

RAD access order

Use the access order section of a RAD to specify the names of and the order in which Netcool Configuration Manager should access specific network protocols. Access order is not called out explicitly in the GUI as it is defined in the XML document; however, it is represented in the GUI by the list of enabled access types.

XML tags example

The XML document explicitly calls out the access order, and declares which access types will be used.

The following example shows the XML tags used in the access order section of a RAD:

```
<access-order>
    <name>ssh</name>
    <name>telnet</name>
    <name>alt-telnet</name>
</access-order>
```

Table 2. XML tags description	
XML tag	**Description**
`<access-order>`	Specifies the access order section of a RAD. The `<access-order>` XML tag is typically followed by one or more `<name>` XML tags. The order in which access types are attempted is specified in a RAD file. So, in this example Netcool Configuration Manager would start with SSH, then TELNET, and so forth. As soon as a successful connection is made, the access type listed next in line is not required and therefore not attempted.
`<name>`	

Tip: GUI access order: Access types are displayed by right clicking on the RAD in the Resource Browser, and selecting **Edit**. The access types displayed in the GUI need to be enabled in the RAD for them to be used. Access types can be enabled by ensuring that the **enabled** checkbox against the access type is selected. An access type that is italicized in the GUI is not enabled in the RAD. The order in which they are listed reflects the order in which the access types are used to access the device. For example, if the access type of 'ssh' is listed before 'telnet', the 'ssh' access type will be used to attempt to connect to the device first. If a successful connection is made to the device using 'ssh', then the 'telnet' access type will be ignored. If the connection is not successful using 'ssh', 'telnet' is then used to attempt connection to the device.

Remember: At this point, the name of the access type should not necessarily reflect the protocol used to connect to the device.

Setting RAD Rollback

Use the rollback section of a RAD to set criteria for subsequent device changes.

Syntax

The following example shows the XML elements used in the rollback section of a RAD. This rollback is for the TELNET protocol that is specified in the access order section of a RAD.

```
<access-order>
    <name>telnet</name>
</access-order>

<rollback-options>

    <option name="NO_ROLLBACK">
        <description>No rollback</description>
        <required>false</required>
        <recommended>false</recommended>
    </option>

    <option name="MODELLED_ROLLBACK">
        <description>Use modelled rollback</description>
        <required>false</required>
        <recommended>true</recommended>
    </option>

    <option name="REBOOT_ROLLBACK">
        <description>Reload the configuration and reboot thedevice.</description>
        <required>false</required>
        <recommended>false</recommended>
    </option>

</rollback-options>
```

Description

The rollback XML elements define which options are available for devices using this RAD. The options apply for all access types. Therefore, there is no need to set one for each access type. That is, you do not need to set one rollback for TELNET, another rollback for SSH, and so forth.

The options define what to do in case of an error while applying a CommandSet or NativeCommandSet. If the configuration was partially changed, ITNCM - Base needs to know what action to take in order to restore the device.

The following table describes each of the XML elements used in the rollback section of a RAD.

XML element	Description
`<rollback-options>`	Specifies the rollback section of a RAD. The `<rollback-options>` XML tag is typically followed by one or more `<option name>` XML tags.
`<option name>`	Specifies the name of a rollback option. Specify one of the following values: • `NO_ROLLBACK` — Leave as is. • `MODELLED_ROLLBACK` — Determines how to restore the device. This feature only applies to devices that use SmartModels and allows an intelligent rollback. • `SPECIAL_ROLLBACK` — Determines how to restore Juniper devices. This feature only applies to devices that use SmartModels and allows an intelligent rollback. • `REBOOT_ROLLBACK` — Pushes the original configuration to the box and reboots the device. Each `<option name>` tag has three options that are specified with the following XML tags: • `<description>` • `<required>` • `<recommended>`
`<description>`	Specifies the text that is displayed in the GUI.
`<required>`	Specifies that the option is forced selected. Specify the value `true` to force select the option. Otherwise, specify the value `false`.
`<recommended>`	Specifies that the option is selected by default, but the user may deselect. Specify the value `true` to select the option by default. Otherwise, specify the value `false`.

Notes

For a Command Set or Native Command Set change, the GUI queries the server to get the rollback options allowed for the device. If the user selects more than one device with different VTMOS, the options at the VT level are selected. This allows the GUI to display more options than a particular device supports. Despite what the user selects from the GUI, ITNCM - Base applies only the options allowed for a particular device, in the order specified by the RAD.

ITNCM - Base attempts the rollback options in the order specified, until one succeeds, or it runs out of options. After the RADs have been completed, the user can specify how far back to rollback the device using the GUI. ITNCM - Base can either rollback to the starting config, or rollback to after the last successful command set.

See the *ITNCM User Guide* for additional information on the help text for information on applying Command Sets or Native Command Sets.

Access types

Use the access types section of a RAD to describe the communication that is used between the network resource and ITNCM - Base.

While access order defines the order of each connection protocol to a device, the access type is where a connection is made to the device. There are various attributes that can be set on an access type to control it's connection to the device, and determine the various actions a UOW can perform once connection is made.

Syntax

The following example shows the XML elements used in the access types section of a RAD. The access type specified in the example is the TELNET protocol.

```
<!-- Begin access types section of a RAD -->
<access-types>

    <!-- Begin access type definition for TELNET protocol -->
    <access-type name="telnet">

            <script-id>default</script-id>
            <ssh-type>none</ssh-type>

            <streaming-put-flag>false</streaming-put-flag>
            <streaming-get-flag>true</streaming-get-flag>

            <native-compare-flag>false</native-compare-flag>
            <import-prewrite-flag>true</import-prewrite-flag>

            <sync-prewrite-flag>true</sync-prewrite-flag>
            <import-report-diffs-flag>false</import-report-diffsflag>

            <user/>
            <password/>
            <enable-password/>
            <prev-user/>
            <alt-user/>
            <prev-password/>
            <alt-password/>

            <prev-enable-password/>
            <alt-enable-password/>

            <hostname/>
            <port>23</port>

            <ssh1>
                <cipher>des</cipher>
            </ssh1>

            <ssh2>
                <cipher>blowfish</cipher>
            </ssh2>

    <!-- End access type definition for TELNET protocol -->
    </access-type>

<!-- End access types section of a RAD -->
</access-types>
```

Description

The following table describes each of the XML elements used in the access types section of a RAD. These XML elements map to characteristics that describe how ITNCM - Base communicates with a device.

Note: Not all of the XML elements described in the table appear in the previous example of the access types section of a RAD.

XML element	Description
<additional-errors>	Enables additional device error tokens to be created. These error tokens are appended to the values in default.errors in the device script.
<alt-enable-password>	Specifies a flag used for authentication.
<alt-password>	Specifies a flag used for authentication.
<alt-enable-password>	Specifies a flag used for authentication.
<alt-user>	Specifies a flag used for authentication.
<configDataType>	Specifies the type of data that ITNCM - Base should get from the device. This option takes one of the following values: • CLI • BIN • CLIandBIN
<configEditPrompt>	Sets the prompt used by a device in config edit mode.
<ConnectResponseTimeout>	Specifies the timeout (in milliseconds) for the connection to be established and the user to successfully login.
<ConnectRetryDelay>	Specifies the time (in milliseconds) between each connection retry, only if the option to use a retry has been selected.
<ConnectionTimeout>	Specifies the timeout (in milliseconds) used when sending the connect prompt to a device. This is used to determine if there is still have an active connection.
<enable-password>	Specifies a flag used for authentication.
<enablePrompt>	Sets the prompt that a device uses in enable mode.
<hostname>	Specifies an override for the hostname associated with this device. This option is also used when setting up out of band management.
<import-char-streaming-flag>	Specifies a flag that instructs ITNCM - Base to send one character at a time while logging into a device.
<import-char-streaming-time-interval>	Specifies how fast in milliseconds to stream the character.
<import-prewrite-flag>	Specifies a flag that instructs ITNCM - Base to write the running config to the stored config on a Device before the device is read in.
<import-report-diffs-flag>	Specifies a flag that instructs ITNCM - Base to perform a deep compare and report the differences on an import.

XML element	Description
<lbl-mode-flag-forncs>	Specifies a flag that enables or disables line by line mode when applying a native command set. Specify the value `true` to enable line by line mode. Otherwise, specify the value `false` to disable line by line mode.
<MaxResponseTimeout>	Specifies the maximum time (in milliseconds) to wait for a response from a device.
<native-compare-flag>	Specifies a flag that instructs ITNCM - Base to do a raw CLI compare first. If the configs are the same then it is done. If not, then the CLI is converted to XML and a deep compare is done. Before the raw CLI compare is done, any lines that are known to be different are removed.
<password>	Specifies a flag used for authentication.
<port>	Specifies the port number used for communication with the device.
<prev-enable-password>	Specifies a flag used for authentication.
<prev-password>	Specifies a flag used for authentication.
<prev-user>	Specifies a flag used for authentication.
<Prompt>	Specifies an option to enable the prompt on the device after login. **Note:** This option is future functionality.
<reboot-on-config-load>	Specifies a flag that instructs ITNCM - Base to reboot the system after a config load. Specify the value `true` to instruct ITNCM - Base to reboot the system after a config load. Otherwise, specify the value `false` to prevent ITNCM - Base from rebooting the system after a config load. The default for this flag is `false`.
<ResponseTimeout>	Specifies the time (in milliseconds) waited for a single wait statement. This is continually reset with reading the buffer and receiving the token. The maximum response timeout will override this if it is affected while retrieving data. Some devices keep sending data and keep resetting the response timeout, so the maximum response time out is used to override this and prevent an infinite loop.
<script-id>	Specifies the name of the device script to use. ITNCM - Base supplies two default values: `default` and `ssh`. The user can create as many scripts as required. If the user creates additional scripts, they will show up in the script section at the bottom of the RAD.
<socketConnectTimeout>	Specifies, in milliseconds, the connect timeout. Use this option to control the socket connection timeout for auto discovery.

XML element	Description
<sourceAddress>	Specifies an address used for source routing when you want to send a different address than the server.
<ssh1>	Specifies a flag used for the encryption method for SSH 1 type connections.
<ssh2>	Specifies a flag used for the encryption method for SSH 2 type connections.
<ssh-type>	Specifies the type of SSH connection (or no SSH connection) to set up. This option takes one of the following values: • SSH — Set up an SSH 1 connection type. • SSH2 — Set up an SSH 2 connection type. • none — Do not set up an SSH connection type.
<streaming-get-flag>	Specifies a flag that instructs ITNCM - Base as to which communication method to use. Data that is retrieved from the device is streamed over the communication path that was used to connect to the device (typically Telnet OR SSH). This is dependent on the access-type. The alternative is to use file transfer methods to post the data to FTP server.
<streaming-put-flag>	Specifies a flag that instructs ITNCM - Base to stream all the changes to the device through streaming TELNET. Data that is sent to the device is streamed over the communication path that was used to connect to the device (typically Telnet OR SSH). This is dependent on the access-type. If the flag is set to `false`, then ftp/tftp is used to apply the changes.
<sync-prewrite-flag>	Specifies a flag that instructs ITNCM - Base to write the running config to the stored config on a Device before a device is read in for a sync.
<Timeout>	Specifies the timeout used for sending a character at a time. If you select the **Character** check box, then this option applies. Some devices require ITNCM - Base to send one character at a time when logging in.
<update-Resource-InfoOn-Config-Change>	Each time a config change occurs, this retrieves the data for all model and device specific data. By default this only happens at import.
<update-Resource-ConfigOn-Config-Change>	Each time a config change occurs, this retrieves the data for all configuration specific data. By disabling this, you will end up with a Stale Configuration because the system made a change to device but never updated the database with the new config data.
<user>	Specifies a flag used for authentication.

File-based access method

Using the file-based access method allows you to obtain device information from a file placed on a server.

The file-based access method obtains device information from a file server. Typically, a customer downloads the device information to the file server, and then Netcool Configuration Manager imports the device information from the ftp/scp server. The file is downloaded to the worker server that is processing the UOW. The worker server connects directly to the file server that contains the device info. Once the information has been retrieved, it is treated just like any other device.

1. In the Netcool Configuration Manager Resource Browser, click **File** > **New** and select a resource type of **Resource Access**.

   Note: VTMOS filters can be applied as with any other Resource Access resource.
2. Right-click the Resource Access, and click **Edit** to display an editing panel for the new Resource Access resource.
3. Click **Add** on the **Access Types** tag to add a new access type called `file`.
4. Select the new **file** access type, and select the following check boxes on the Transport tab:
   - **Enabled**
   - **Streaming: Put**
   - **Streaming: Get**
5. On the SSH Options tab, select the **ssh2** option from the SSH Type selection menu.
6. Select a SSH1 Cipher of **des3**, and a SSH2 cipher of **aes128**.
7. On the Command Line tab, select **file** from the Script menu.

   Note: If the **file** option is not available in the Script menu, type `file` in the **Script** text field.
8. On the Configuration tab, select the check boxes for **Native Compare** and **Allow Line by Line for Native Commend Set**.
9. Enter a Config data Type of `CLI`.
10. Select the **Scripts** tab, and then click **Add** to create a new script with the name `file`.
11. Enter the required script in the text window, for example:

```
#turn on flags to get info from file server
getConfigFileServer=true
getModelFileServer=true
getDiagFileServer=true
getVersionFileServer=true
getBinaryFileServer=true
putConfigFileServer=true

### Defaults for sending commands. Errors must be separated by , not spaces
default.prompt=$
log-in.prompt=$
default.error=Error,Invalid,Incomplete command
default.errorResponse=Error sending command

### Connection Global
# if connect.* is not present use connect.all.properties
connect.errorResponse=Unable to connect to device
connect.09.wait=$

# check for Running config and stored config values multipleConfigs or SingleConfig
config.check.end=singleConfig

# Signals start of config
config.start=!
# Signals end of config
config.end=\nend\r

# Identifies error retreiving config.  Must be separated by ,
config.fail=Error,Invalid,Incomplete command

# Info
# these commands are used to gain some information on the hardware installed
# in the device
```

```
diag.01.setReturnBuff=test#
diag.end=$

# Model
## using the command below, or similar, select the text that will allow
determination of a model
model.01.setReturnBuff=EBS-NSP-6.1
model.end=$
model.FIND-BEGIN=
model.FIND-END=

# Version
## using the command below, or similar, select the text that will allow determination
## of an OS version
config.version.setReturnBuff=<your version value goes here>
config.version.end=$
config.version.FIND-BEGIN=
config.version.FIND-END=
config.version.FEATURE-FIND-BEGIN=Software (
config.version.FEATURE-FIND-END=),

# Running config
###
## The setFtpFileName parameter can be used to specify the file containing the
## configuration content by its exact name if needed.
## To use this parameter, uncomment the following line and supply the file name.
## If you use this parameter, comment out the line for getFtpFileName.
###
#config.running.01.setFtpFileName=<filename>
###
## Use the getFtpFileName parameter to identify the file based on the
## name/address of the device.
## The addressName parameter will use the name of the resource. If you modify
## the directory structure on the source file server, then there is one edit to
## the device script required. The "f" parameter in the "cut" operation needs to
## be incremented to match the number of directory levels. If you add a level
## on the source file server, then "-f4" is changed to "-f5" and so on.
## The other change required is an addition of a corresponding directory to the
## /home/icosftp directory to match the source directory. This is required due to
## the way the file-based access method uses the ftp resource.
###
## The end parameter, either first or last, specifies the first or last instance
## of the file based upon the order of the file listing used earlier in the command:
## ls -t lists files with the newest (most recently modified file) listed first.
## ls -tr lists files with oldest (least recently modified file) listed first.
###
config.running.01.getFtpFileName=ls -t $ftpPath$/*$addressName$* | cut -d"/" -f4,first
config.running.02.getSCP=$ftp_username$:$ftp_password$@$ftp_hostname$:$ftpPath$/
$ftp_filename$
config.running.end=$
config.running.FIND-BEGIN=# extended LDIF
config.running.FIND-END=numEntries:

# Stored config
config.stored.send=show startup-config\r
config.stored.end=\nend\r
config.stored.FIND-BEGIN=version
config.stored.FIND-END=\nend\r

### Sends a change back to the file server
ftp.01.send=putSCP://$ftp_username$:$ftp_password$@$ftp_hostname$/
$ftp_filename$ running-config\r
ftp.09.wait=$
```

12. Now import the device.

NCM file-based driver setup using a single server

Fix Pack 2

Overview

You can import devices using the file-based driver access method from a separate server, or you can import files (devices) from the same worker server. Using a two server approach is typical in a production environment where network policies prevent direct interaction between the Netcool Configuration Manager worker servers and the network EMS or nodes. Typically, this file server resides in a DMZ where the production devices can reach one NIC on the server while Netcool Configuration Manager can

reach a separate NIC. In other situations files can be located on the same worker server and Netcool Configuration Manager transfers the file from one directory to another.

Single server implementation

You need to create a realm from the Netcool Configuration Manager UI, and within that realm you must create an authentication resource, a file transfer resource, and any network resources that will be created and imported. On the worker server that will be performing the imports, a separate directory must be created to store the files that will be imported (source files). These source files will then be transferred on the server to the destination directory specified in the File Transfer Resource.

Detailed setup process

Perform the following steps on the worker server.

Note: This example uses the following default values:

User
icosftp

User's home directory
/home/icosftp

Realm
ITNCM/single-server

1. Create a file directory on the worker server named `/home/icosftp/single`. This directory specifies the source directory for the files. Place any device configuration files in this directory ensuring that they have read permissions set as appropriate for the icosftp user.
2. Create a File Transfer Resource with the following parameters (this FTP resource specifies the destination directory for the files):

   Name
   ftpInfo

   Host
   localhost

   Username
   icosftp

   Password
   icosftp user's password

   Path
   /home/icosftp

   Passive Mode
   checked
3. Create an Authentication Resource for the default FTP user, if it does not already exist. The following example settings use the default `icosftp` user:

   Username
   icosftp

   Password
   icosftp user's password

   Enable Password
   leave blank

   Retry Count
   0 (zero)

   Retry Delay
   0 (zero)

   Ignore
   False

4. Create a Network resource specifying a Name, Vendor, Type, Model, and OS.
5. Either edit the Resource Access on the Network Resource (right-click on the **Network Resource** and select **Resource Access**) or create a Resource Access (click on **File** > **New** > **Resource Access**).

   Note:

   If you create a new Resource Access
   You add the file script to the resource access.

   Follow the steps described in the File-based access method topic starting with Step 3.

   If you edit the Resource Access on the network resource
   Modify the file script.

   Follow the steps described in the File-based access method topic starting with Step 10.
6. Edit the file script in the Resource Access Document (RAD) to specify the source directory for the files, as shown in the following sample.

```
config.running.01.getFtpFileName=ls -t $ftpPath$/ single /*$addressName$* | cut -d"/" -f5 ,first
config.running.02.getSCP=$ftp_username$:$ftp_password$@$ftp_hostname$:$ftpPath$/single/$ftp_filename$
config.running.end=$
```

   Note: Note the introduction of the directory name `single` into the path, and the change to the directory depth to 5 (was 4), highlighted.
7. Import the network device.

Writing changes to a named configuration file

You can use the NSM rest API (but not the GUI) to write or delete full device configurations or full device service configurations to named files and directories for devices to which you have file-based access. A service is a subset of the full configuration of a device. You can write and delete either device configurations or device service configurations to any one device, not both.

Note: To use this feature, you need the appropriate device driver, released after Netcool Configuration Manager V6.4.2 fix pack 8. To find out if a particular updated device driver is available, contact IBM Support.

Creating directory paths and configuration files on local and remote systems

To create a directory path and configuration file, post a service by using the NSM REST API. In the service template, supply a directory path and file name in the `CUSTOM_UOW_LABEL_1` `uowParameter` in the service template. For example, `MAC999999/SN77777_1.xml`.

The directory and file that you specified are created on the local file system under the `$ftpPath$/$addressName$` directory.

The configuration file contains the full configuration for a device or a device service. The directory and filename are created on the remote system under the `$ftp_altpath$` directory. The file contains the full configuration for a device or device service.

Valid characters for directory and file names are alphanumeric characters and the following characters:

```
-
_
.
/ (for path separator only)
```

No two devices can share the same user-supplied path.

If you use a device script to create an initial configuration file for service configuration updates, ensure that the Service Templates or Command sets used at posting of the service contain a full service configuration. Do not use `$addressName$.cfg` as the user-supplied file name, unless the user-supplied device script can handle it.

If you use a device script to create an initial configuration file for device configuration updates, ensure that the Service Templates or Command sets used at posting of the service contain a full device

configuration. Each post of a service for a device must use the same directory path and configuration file name.

Importing full configurations

You must write a script that creates an initial configuration file `$ftpPath$/$addressNam $addressName$.cfg` with a dummy value.

For device configuration updates, run this script at the first import after creation, and not for subsequent service posts or configuration imports.

For service configuration updates, run this script at the first import after creation. On subsequent imports the script also aggregates each of the 'service' configuration files into the device configuration file `$addressName$.cfg` in the `$ftpPath$/$addressName$` directory. This file is used for the device import operation.

Deleting configuration files associated with a service or device

The following considerations apply to both local and remote systems.

Deleting a service using the REST `Delete` and the service ID of the posted service deletes the configuration files that were associated with the original posting of the service.

Deleting configuration files associated with a device deletes the full configuration, because the full configuration was supplied with the original service post operation.

Deleting device configuration files when a device is deleted

The following considerations apply to both local and remote systems.

To delete the configuration files for a device at the same time as the device is deleted, complete the following steps:

1. Post a service that removes a device, and supply the value `delete_device_file_based_configs` in the Service template `CUSTOM_UOW_LABEL_1` `uowParameter`
2. Supply the `delete_device_file_based_configs` value.

The device is removed, and the `$ftpPath$/$addressName$` directory and all its contained files and subdirectories are deleted from the local file system.

If any of the files or subdirectories under `$ftpPath$/$addressName$` also exist under `$ftp_altpath$` on the remote system, they are also deleted from that location.

Restriction: Before continuing, you must configure Worker servers so that a single worker handles all the UOWs asociated with a particular filed-based access device. The ftp resource associated with the device must point to the worker server host.

Restriction: Device renaming is not supported.

To write changes to a named configuration file, complete the following steps.

1. Set up file-based access as described above.
2. Create an Authentication Resource for the default FTP user, if it does not already exist. The following example settings use the default `icosftp` user:

   Username
   icosftp

   Password
   icosftp user’s password

   Enable Password
   leave blank

   Retry Count
   0 (zero)

   Retry Delay
   0 (zero)

Ignore
False

3. Create an FTP Resource to specify the destination directory for the files.
   a. Add a new entry and set the following parameters:
      - Name: ftpInfo
      - Host: localhost
      - Username: icosftp
      - Password: password of icosftp user
      - Path: /home/icosftp
      - Passive Mode: unchecked
   b. Add another entry and set the following parameters:
      - Name: altftpInfo
      - Host: address of remote host
      - Username: icosftp
      - Password: password of icosftp user
      - Path: /home/icosftp
      - Passive Mode: unchecked
4. Create a custom UOW label to use for the custom file name.
   a. Select **Tools** > **System Properties** from the Systems Manager.
   b. Set the Custom UOW label 1 property to Config_Dir_Name.
   c. Set the Custom UOW label 1 state to have the value Optional.
5. Complete either this step, Step "5" on page 70, for device configuration updates, or Step "6" on page 71 for service configuration updates.
   a. Write a script that generates a new configuration file. The script is run from the Resource Access device script, as shown in Step "5.b" on page 70. The script must achieve the following results:
      - If you want the device to be given an initial configuration after it is first created, the script must create a new configuration file using the directory and filename that are passed to the script, if the file does not exist already, and then populate the file.
      - Do not create or populate the configuration file if it exists already for the device.
      - Do not return any data. Returned data is added to the configuration.
      - Change the permissions on the initial configuration file for the device using the `chmod 777` command. The script is usually run as the `icosuser`, whereas FTP is usually run as the `icosftp` user.
   b. Edit the Resource Access that you created previously for file-based device access, select the **file** access type, and make the following changes:
      i) Uncheck **Streaming: Put** in the **Transport** tab.
      ii) Check **Update on change** in the **Configuration** tab.
      iii) Edit the device script and make the following changes:
         a) In the `# Device script properties` section, change `putConfigFileServer` to `putConfigFileServer=configured_for_remote_scp`.
         b) Add a new Device script `removeConfigDir` section:

```
# New section in device script, removes all config files and directories
 for a device, both locally and remotely.
removeConfigDir.01.getDeviceFileDirContent=ls $ftpPath$/$addressName$/
 | cut -d"/" -f5,all
removeConfigDir.02.removeRemoteContent=$ftp_altusername$:
$ftp_altpassword$@$ftp_althostname$:$ftp_altpath$/
```

```
$device_file_dir_content$
removeConfigDir.03.exec=rm -rf $ftpPath$/$addressName$\r
```

c) Add a new Device script `removeConfigFile` section:

```
# New 'removeConfigFile' section.
# Removes a config file for a device, both locally and remotely, this will
be triggered as a result of a 'delete service' being performed.
# 'Update on Change' should be set in the Resource access, so that changes
here will be reflected in the configuration.
removeConfigFile.01.removeRemoteContent=$ftp_altusername$:$ftp_altpassword$
@$ftp_althostname$:$ftp_altpath$/$config_dir$/$ftp_filename$
removeConfigFile.02.exec=rm -rf $ftpPath$/$addressName$/$config_dir$/
$ftp_filename$\r
removeConfigFile.03.exec=rm -rf $ftpPath$/$addressName$/$addressName$.cfg\r
```

d) In the Device script `config.running` section, replace `config.running.01` and `config.running.02` in the file-based example with the following lines:

```
config.running.01.exec=mkdir -p $ftpPath$/$addressName$\r
config.running.02.exec=$ftpPath$/telus_script_for_initial_config.sh -d
$ftpPath$/$addressName$ -f $addressName$.cfg\r
config.running.03.setFileName=$addressName$.cfg
config.running.04.setSubDirForFtpPath=$addressName$
config.running.05.readSCPFile=$ftpPath$/$subDirForFtpPath$/$ftpFileName$
```

Note: The line beginning `config.running.02.exec` is optional. It is provided as an example of how to set up an initial default configuration that takes effect at first import after creating a device. Replace the script name `script_for_initial_config.sh` with the script that you created in this step. The location of the script, and the argument names `-d` and `-f` in the preceding code can be changed.

iv) If you have a `cp -i` alias set up on the host, undefine the alias by using `unalias cp -i`, otherwise `cp -f` does not work in the following step.

v) In the Device script `ftp` section, replace the entire `ftp` section in the example with the following lines:

```
ftp.01.exec=mkdir -p $ftpPath$/$addressName$/$config_dir$/\r
ftp.02.exec=mv -f $ftpPath$/$ftp_filename$ $ftpPath$/$addressName$/
$config_dir$/\r
ftp.03.exec=cp -f $ftpPath$/$addressName$/$config_dir$/$ftp_filename$
 $ftpPath$/$addressName$/$addressName$.cfg\r
ftp.04.setSubDirForFtpPath=$addressName$/$config_dir$
ftp.05.scpFile=$ftp_altusername$:$ftp_altpassword$@$ftp_althostname$:
$ftp_altpath$/$config_dir$/r
```

6. Complete either this step, Step “6” on page 71, for service configuration updates, or Step “5” on page 70 for device configuration updates.

a. Write a script that generates a new configuration file. The script is run from the Resource Access device script. The script must achieve the following results:

- If you want the device to be given an initial configuration after it is first created, the script must create a new configuration file using the directory and filename that are passed to the script, if the file does not exist already, and then populate the file.
- Aggregate any service configuration files present in the directory `$ftpPath$/$addressName$` up to a single device configuration file for import: `$addressName$.cfg`.
- Handle any specific ordering or formatting requirements for aggregating the individual service configuration files up to a device configuration file.
- Do not return any data. Returned data is added to the configuration.
- Change the permissions on the initial configuration file for the device using the `chmod 777` command. The script is usually run as the `icosuser`, whereas FTP is usually run as the `icosftp` user.

b. Edit the Resource Access that you created previously for file-based device access, select the **file** access type, and make the following changes:

i) Uncheck **Streaming: Put** in the **Transport** tab.

ii) Check **Update on change** in the **Configuration** tab.

iii) Edit the device script and make the following changes:

a) In the `# Device script properties` section, change `putConfigFileServer` to `putConfigFileServer=configured_for_remote_scp`.

b) Add a new Device script `removeConfigDir` section:

```
# New section in device script, removes all config files and directories
 for a device, both locally and remotely.
removeConfigDir.01.getDeviceFileDirContent=ls $ftpPath$/$addressName$/
 | cut -d"/" -f5,all
removeConfigDir.02.removeRemoteContent=$ftp_altusername$:
$ftp_altpassword$@$ftp_althostname$:$ftp_altpath$/
$device_file_dir_content$
removeConfigDir.03.exec=rm -rf $ftpPath$/$addressName$\r
```

c) Add a new Device script `removeConfigFile` section:

```
# New 'removeConfigFile' section.
# Removes a config file for a device, both locally and remotely, this will
be triggered as a result of a 'delete service' being performed.
# 'Update on Change' should be set in the Resource access, so that changes
here will be reflected in the configuration.
removeConfigFile.01.removeRemoteContent=$ftp_altusername$:$ftp_altpassword$
@$ftp_althostname$:$ftp_altpath$/$config_dir$/$ftp_filename$
removeConfigFile.02.exec=rm -rf $ftpPath$/$addressName$/$config_dir$/
$ftp_filename$\r
removeConfigFile.03.exec=rm -rf $ftpPath$/$addressName$/$addressName$.cfg\r
```

d) In the Device script `config.running` section, replace `config.running.01` and `config.running.02` in the file-based example with the following lines:

```
config.running.01.exec=mkdir -p $ftpPath$/$addressName$\r
config.running.02.exec=$ftpPath$/telus_script_for_initial_config.sh -d
$ftpPath$/$addressName$ -f $addressName$.cfg\r
config.running.03.setFileName=$addressName$.cfg
config.running.04.setSubDirForFtpPath=$addressName$
config.running.05.readSCPFile=$ftpPath$/$subDirForFtpPath$/$ftpFileName$
```

Note: The line beginning `config.running.02.exec` is optional. It is provided as an example of how to set up an initial default configuration that takes effect at first import after creating a device. Replace the script name `script_for_initial_config.sh` with the script that you created in this step. The location of the script, and the argument names `-d` and `-f` in the preceding code can be changed.

iv) If you have a `cp -i` alias set up on the host, undefine the alias by using `unalias cp -i`, otherwise `cp -f` does not work in the following step.

v) In the Device script `ftp` section, replace the entire `ftp` section in the example with the following lines:

```
ftp.01.exec=mkdir -p $ftpPath$/$addressName$/$config_dir$/\r
ftp.02.exec=mv -f $ftpPath$/$ftp_filename$ $ftpPath$/$addressName$/
$config_dir$/\r
ftp.03.exec=cp -f $ftpPath$/$addressName$/$config_dir$/$ftp_filename$
 $ftpPath$/$addressName$/$addressName$.cfg\r
ftp.04.setSubDirForFtpPath=$addressName$/$config_dir$
ftp.05.scpFile=$ftp_altusername$:$ftp_altpassword$@$ftp_althostname$:
$ftp_altpath$/$config_dir$/r
```

7. Edit a service template to include the new parameter `uowParameter` with name `CUSTOM_UOW_LABEL_1`. Adding the parameter allows the configuration file name to be passed in REST calls. The following example service template shows a command set that adds snmp contact and snmp location elements in a configuration, with the new parameter added:

```
<serviceTemplate name="SNMP_test" description="NSM Service Template to add
 and delete SNMP info">
    <clientParameters>
```

```
        <clientParameter>
            <name>LOCATION</name>
        </clientParameter>
        <clientParameter>
            <name>CONTACT</name>
        </clientParameter>
    </clientParameters>
    <uowParameters>
        <uowParameter>
            <name>CUSTOM_UOW_LABEL_1</name>
            <description>The configuration file name</description>
        </uowParameter>
    </uowParameters>
    <implementations>
        <implementation>
            <rules>
                <rule type="DeviceType">
                    <ruleProperty name="Vendor" value="mediatrix"/>
                    <ruleProperty name="Type" value=".*"/>
                    <ruleProperty name="Model" value=".*"/>
                    <ruleProperty name="OS" value=".*"/>
                </rule>
            </rules>
            <serviceOperations>
                <serviceOperation type="CREATE">
                    <operations>
                        <operation name="ITNCM/mediatrix/SNMP-add" type="COMMANDSET">
                            <parameters>
                                <parameter name="LOCATION"/>
                                <parameter name="CONTACT"/>
                            </parameters>
                        </operation>
                    </operations>
                </serviceOperation>
                <serviceOperation type="DELETE">
                    <operations>
                        <operation name="ITNCM/mediatrix/SNMP-delete" type="COMMANDSET">
                            <parameters>
                                <parameter name="LOCATION"/>
                                <parameter name="CONTACT"/>
                            </parameters>
                        </operation>
                    </operations>
                </serviceOperation>
            </serviceOperations>
        </implementation>
    </implementations>
</serviceTemplate>
```

The following example service template removes a device, and has the new parameter added:

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceTemplate name="remove_device" description="To remove devices">
    <uowParameters>
        <uowParameter>
            <name>CUSTOM_UOW_LABEL_1</name>
            <description>The configuration file name</description>
        </uowParameter>
    </uowParameters>
    <implementations>
        <implementation>
            <rules>
                <rule type="DeviceType">
                    <ruleProperty name="Vendor" value="Mediatrix"/>
                    <ruleProperty name="Type" value=".*"/>
                    <ruleProperty name="Model" value=".*"/>
                    <ruleProperty name="OS" value=".*"/>
                </rule>
            </rules>
            <serviceOperations>
                <serviceOperation type="CREATE">
                    <operations>
                        <operation name="REMOVE" type="REMOVE">
                        </operation>
                    </operations>
                </serviceOperation>
                <serviceOperation type="DELETE">
                    <operations>
                        <operation name="REMOVE" type="REMOVE">
                        </operation>
```

```
                    </operations>
                </serviceOperation>
            </serviceOperations>
        </implementation>
    </implementations>
</serviceTemplate>
```

8. When you submit a configuration update by using the GUI, enter the custom name for the configuration file in the **Config_Dir_Name** field on the **Describe Work** screen.
9. The following example NSM service post body defines the configuration file name as `MAC999999/SN77777_1.xml`.

   XML example:

```
<serviceTemplate id="101">
    <deviceID>402</deviceID>
  <uowParameters>
    <uowParameter>
      <name>CUSTOM_UOW_LABEL_1</name>
      <value>mediatrix_device.cfg</value>
    </uowParameter>
  </uowParameters>
    <clientParameters>
        <clientParameter>
            <name>LOCATION</name>
            <value>The Shire</value>
        </clientParameter>
        <clientParameter>
            <name>CONTACT</name>
            <value>Bilbo Baggins</value>
        </clientParameter>
    </clientParameters>
</serviceTemplate>
```

   JSON example:

```
{
       "serviceTemplateId" : "101",
       "deviceID": "402",
       "clientParameters": [
      {
         "name": "LOCATION",
         "value": "The Shire"
      },
      {
         "name": "CONTACT",
         "value": "Bilbo Baggins"
      }
   ],
   "uowParameters": [
     {
         "name": "CUSTOM_UOW_LABEL_1",
         "value": "MAC999999/SN77777_1.xml"
      }
   ]
}
```

10. The following example NSM service post body for removing a device has the `delete_device_file_based_configs` value that triggers the deletion of configuration files associated with the device.

    XML example:

```
 <serviceTemplate id="105">
    <deviceID>3207</deviceID>
  <uowParameters>
    <uowParameter>
      <name>CUSTOM_UOW_LABEL_1</name>
      <value>delete_device_file_based_configs</value>
    </uowParameter>
  </uowParameters>
</serviceTemplate>
```

JSON example

```
{
        "id" : "105",
        "deviceID": "3207",
    "uowParameters": [
      {
          "name": "CUSTOM_UOW_LABEL_1",
          "value": "delete_device_file_based_configs"
       }
    ]
}
```

Editing the resource access doc with XML

Use XML to edit the resource access doc (RAD).

Consider creating a stand-alone RAD for each VTM (Vendor, Type, Model) rather than a device level RAD due to the maintenance cost.

The following steps describe how to edit the RAD using XML.

1. Access the RAD dialog.
2. Click XML in the upper right corner of the dialog. The XML form is displayed.
3. Make changes to the XML as required, for example modifying the device *<sourceAddress>*.
4. Click File | Save, or click on the Save icon.

File transfer

When ITNCM - Base is working with a device, it can be setup to use streaming to get information from the device, and to make changes to the device.

File transfer can be used by setting the different flags in the RAD. ITNCM - Base will allow the set up of almost any type of configuration that is required.

The following is an example of how FTP mode works.

1. Log into a device and send commands to the device to send information to, or get information from ITNCM - Base. The FileTransfer GR is used for information about what ftp/tftp server to use. The server can reside on the same system as ITNCM - Base or not.
2. ITNCM - Base locates the FileTransfer GR by looking into the same realm as the device, then moving up the tree until the top realm is reached.
3. If a FileTransfer GR is not found then ITNCM - Base retrieves the required information from the database.
4. If the file transfer server is a different server, ITNCM - Base moves the file as needed to get the information from ITNCM - Base to the file transfer server and then to the device.

The following is an example of a FileTransfer GR.

```
<?xml version="1.0" encoding="UTF-8"?><ftp encrypted="false">
<!--
<entry>
<name>ftpInfo</name>
<host>ftphost.example.com</host>
<username>icosuser</username>
<password value="current">foo</password>
<password value="previous">bar</password>
<password value="alt">foo</password>
<path>/pub/change.me</path>
<mode>active</mode>
</entry>
-->
<!--
<entry>
<name>altFtpInfo</name>
<host>ftphost.example.net</host>
<username>icosuser</username>
```

```
<password value="current">foo</password>
<mode>active</mode>
</entry>
-->
```

Note: `ftpInfo` and **`altFtpInfo`** are the only two names that can be given to an FTP resource. The GUI may accept other names, but they will not function correctly.

Write access file transfer files

In order for a device to put a file via TFTP on a server, there must already be a writable file of the same name on the server. To create this file, put the file there using FTP. However, since most FTP servers make the files users put there readable but not writable, the file transfer server configuration must be changed as follows, on Linux.

1. Add the following line to `/etc/vsftpd.conf`.
2. `local_umask=000`
3. Restart the FTP daemon or have `xinetd` reload its configuration.

Chapter 5. Custom Drivers

Netcool Configuration Manager drivers encapsulate all vendor-specific operations and provide a layer of abstraction that allows Netcool Configuration Manager to remain vendor agnostic. For example, drivers enable Netcool Configuration Manager to communicate with the different devices used within your network, retrieve configuration, make configuration changes and so on.

Netcool Configuration Manager supports the following types of drivers:

- SmartModel Driver — Specifies a driver that IBM creates and delivers to customers that provides a full modelling of a specific device.
- Standard Mode Driver — Specifies a driver that IBM creates and delivers to customers that provides a limited set of features for manipulating only native configurations.
- Custom Driver — Specifies a driver that a customer creates on site from a supplied IBM template.

Related concepts

Custom driver may add leading 0x0a to password when communicating with devices
If Netcool Configuration Manager appends a leading 0x0A to the password, thereby causing connectivity issues, edit he deviceComInterface.txt as described in this workaround.

Custom driver capabilities

Custom drivers (and standard drivers) differ in functionality from SmartModel drivers, which is described here.

Table 3. Custom driver capabilities. The 'Custom driver capabilities' table lists the differences between custom drivers (and standard drivers), and SmartModel drivers.

SmartModel drivers functionality	**Supported in Standard/Custom Mode?**
AutoDiscovery	Yes, however the AutoDiscovery XML files will need to be customized.
Command Set UOWs	No
Create or Edit Command Sets	No
Driver Update	Yes
File-based Access	Yes
Golden Configurations	No
Import Configuration	Yes
Native Command Sets	Yes
OS Upgrade	Yes
Sync from Device to Netcool Configuration Manager	Yes, however the RAD must have Native Differences enabled.
Sync from Netcool Configuration Manager to Device	No
Report Only Sync	Yes, however the RAD must have Native Differences enabled.
Search Sets	No
Security Sets	No

Table 3. Custom driver capabilities. The 'Custom driver capabilities' table lists the differences between custom drivers (and standard drivers), and SmartModel drivers. *(continued)*	
SmartModel drivers functionality	**Supported in Standard/Custom Mode?**
Submit Configuration, 'Disaster Recovery' and 'Use Raw CLI' options not selected	No
Submit Current Configuration, 'Disaster Recovery' and 'Use Raw CLI' selected	No, but you can achieve the same functionality by importing the device again to cause the Current configuration to become a Versioned Configuration, and then submitting it.
Submit Draft Configuration, 'Disaster Recovery' and 'Use Raw CLI' selected	No, but you can achieve the same functionality by copying the configuration into a Native Command Set and submitting the Native Command Set.
Submit Stale Configuration, 'Disaster Recovery' and 'Use Raw CLI' selected	Yes
Submit Versioned Configuration, 'Disaster Recovery' and 'Use Raw CLI' selected	Yes
View Modelled Configuration	No
View Modelled Differences	No
View Native Configuration	Yes
View Native Differences	Yes

Note: The Netcool Configuration Manager UI may not block end-users from attempting to access the function even though it is not supported by the Standard/Custom driver.

Netcool Configuration Manager - Compliance does not make direct use of drivers but is indirectly affected by the use of Standard mode:

- When in Standard mode, only native policies will give valid results. Modelled policies can be run but will all fail as there is no modelled configuration.
- Since modelled command sets are not supported in Standard mode, compliance cannot execute them as the result of a policy evaluation.

Related reference

Managing device drivers (driverTools.sh)
Use the `driverTools.sh` system script to manage device drivers from the command line.

Related information

Resource access documents (RADs)
The resource access documents (RADs) sets up the communication information between the Worker server and actual devices on the network. They define the protocol and all the connections to be used.

Device characteristics

A device that is a good candidate for support via custom drivers will fit one of the following patterns:

- Created using command line interface via Telnet or SSH.
- Will have relatively short commands of no more than a few hundred characters per line.
- Symmetry of configuration commands, that is the configuration retrieved from the device is itself a list of commands that can be returned to the device.

OR

- or a TL1-based interface

Versioning and optimality

New drivers are automatically versioned. After creating new custom drivers, existing devices are marked to indicate a newer version of the driver has been created.

Devices which had been using the old driver configuration will have the non-optimal yellow arrow icon against them, which indicates that they need a driver update. This is performed in the driver management screens. The driver update moves the device to the latest optimal version. The UOW log also contains information to indicate if the current driver is non-optimal.

If more than one driver is created with the same VTMOS, the earlier one is overriden and upversioned. The one with highest version is always searched for, and used.

The optimality check is performed when assigning a driver to a device at import time.

Custom drivers provide the ability to support new kinds of network device with Standard level functionality. A new custom driver is created using an existing custom or IBM driver as a template. When creating a custom driver you must specify:

1. VTMOS and supported model/OS
2. RAD
3. Device Script

VTMOS & Supported Model/OS

The Vendor/Type/Model/OS (VTMOS), e.g. Cisco/Router/3745/12* for the custom driver is a label that indicates which kinds of device this driver supports in a human-readable fashion. It is specified during the creation of the custom driver. In addition, you must also specify the Supported Models and OSs.

The Supported Models and OSs are used to identity the specific models and OSs that are supported by this driver. Supported models and OSs are globs that are pattern matched against the actual model and OS reported by the device. A driver is only deemed to be applicable to a device if the supported and actual model/OS match.

The following shows the device section responsible for retrieving the actual Model:

```
# Model
 model.modelMaxSize=3000
 model.send=show chassis\r
 model.end=#
 model.FIND-BEGIN=Type :
 model.FIND-END=\r
```

The following shows the device section responsible for retrieving the actual OS (a.k.a “config version”):

```
# Version
 config.version.send=show version\r
 config.version.end=#
 config.version.FIND-BEGIN=TiMOS-
 config.version.FIND-END=
```

You must ensure that the device script is configured such that the actual model and OS strings retrieved from the device correspond to supported Models and OSs specified in the driver. This relationship between actual model/OS and supported model/OS is necessary to allow Netcool Configuration Manager to determine that a given driver is compatible with a given device.

Resource access documents (RADs)

The resource access documents (RADs) sets up the communication information between the Worker server and actual devices on the network. They define the protocol and all the connections to be used.

Editing a RAD allows you to change the protocol used for communication, individual protocol detail, such as SSH cipher, as well as the access order for the protocols. The ports, time-outs chosen and other

Netcool Configuration Manager specifics, for example if the configuration is streamed, are editable in the RAD.

After creating a Custom Driver, you will also need to create a Resource Access Document with the same VTMOS so that Netcool Configuration Manager is able to communicate with the device to retrieve its configuration, send Native Command Sets, and other activities. For this task it is best to study the RAD of a similar device, once you are familiar with the RADs documentation. To view the information you need to customize the Device Script contained within a RAD so that it can connect to the device and retrieve its data, see “Device scripts” on page 44.

When it comes to actually creating the new RAD for the device (using **File** > **New** > **Resource Access**), you can begin by copying the RAD of the similar device:

1. Create a RAD for the Custom Driver.
2. Open the RAD for the similar device, select XML in the top-right corner, select **all** the text you're presented with, and copy that into your clipboard.
3. Open the RAD for Custom Driver device, select XML in the top-right corner, select **all** the text and overwrite it by pasting from your clipboard.
4. Save the Custom Driver RAD.
5. Review all the options in the Custom Driver RAD to ensure they are appropriate, in particular ensure that the Native Compare option in the Custom Driver RAD is selected.

Creating the RAD is an iterative process and can take some time. While developing a RAD, also enable 'FINEST' logging so that you can see all the commands sent by Netcool Configuration Manager and the responses from the device. For more information, see the IBM technote at the following location: http://www-01.ibm.com/support/docview.wss?uid=swg21650723

Related concepts

Resource Access Doc

The Resource Access Doc (RAD) sets up the communication information between the worker server and the actual device. The RAD is one of the most important data structures needed for device interaction. The RAD controls the device connection for IDT and UOWs. ITNCM - Base locates just one RAD when setting up the communication.

Related reference

Custom driver capabilities

Custom drivers (and standard drivers) differ in functionality from SmartModel drivers, which is described here.

Device Scripts

Device scripts are fundamental in communicating with the custom driver. Device scripts are used to manage the commands sent to the device, and retrieve the response.

A device script consists of the following items:

- Device script commands — Each device script contains a list of the commands that will be sent to the device to determine actual model/OS, retrieve or send configurations.
- Device script sections — The required section of the device script performs the functions needed. For example, the default.error section lists string errors. Each section of the device script specifies the commands in the order that they will be executed.
- Device script variables — Each device script makes use of variables, e.g, `$connect_username$`, `$alt_username$`, `$connect_password$`, and so forth.

The `<script-id>` element, specified in the RAD, identifies the device script that Netcool Configuration Manager uses. Multiple device scripts (each with their own name) may be included within the RAD. However, only the one being referenced will be used. By convention, specifying the name default within the `<script-id>` element, means the default device script stored in the database for that device (based on VTMOS) will be used.

The driver device script is available in either text or form based. This is selected at the top right-hand corner when the driver device script is open.

For more information on the RAD and device script, see the *IBM Tivoli Netcool Configuration Manager Administration Guide* .

Driver lifecycle

You can use the Systems Manager to access, manage, and control multiple servers and drivers. Your window displays only those system resources for which you have authorization for display or control.

The Systems Manager is only accessible if the user is a member of a group with the View System activity. Any user wishing to make changes to System Manager must belong to a group with the Manage System activity.

Use this procedure to access the Drivers component within Systems Manager.

1. Select the Systems Manager in the navigation tree.
2. The Systems Manager displays.
3. Navigate to the Drivers component.
4. All drivers available on the system display.
5. Use the following table as a guide to understand the information displayed:

Field Name	Description
UUID	Unique ID to identify the driver.
Vendor/Type/Model/OS	VTMOS of the device associated with the driver.
Version	Driver versions are date based
In Use	Those drivers "In Use" will be denoted by a green checkmark. This indicates that it has been created, or imported.
Status	Indicates the state of the driver. Options are: active and inactive.
Created Date	Date the driver was created.
Custom Driver	Used to differentiate between IBM drivers and custom drivers. Values available are: True and False.
Display name	Alias.

Related concepts

Custom driver may add leading 0x0a to password when communicating with devices
If Netcool Configuration Manager appends a leading 0x0A to the password, thereby causing connectivity issues, edit he deviceComInterface.txt as described in this workaround.

Creating a custom driver

Custom drivers provide the ability to support new kinds of network devices with Standard level functionality. You create a new custom driver by using an existing custom or IBM driver as a template. You then customize the VTMOS, Resource Access Doc (RAD), and device script as required. Newly created custom drivers are automatically distributed to all presentation and worker servers.

The user must be a member of a group who has the "View System" activity. This enables the user to see the Systems Manager in the UI.

You should also understand how to work with Device Scripts and Resource Access Documents (RADs) before creating a custom driver. See Related concepts and Related tasks for information about Device Scripts and RADs.

Use this procedure to create a new custom driver.

1. Navigate to the Systems Manager, and select the Drivers component.
2. From the menu bar, select **Tools** > **New Driver**
3. The **Choose Template** window displays. Using the following table as a guide, select a row in the table to act as the template from which the new custom driver is created. Optionally, the filter drop downs can be used to narrow the number of rows visible in the table. Select **Next** to proceed.

Table 4. Select base driver	
Selection	**Description**
Driver Type	You will be asked to choose which type of driver you want to use as the base driver (template): custom driver or IBM driver.
Custom Drivers	Choose this option if you want to use an existing custom driver you have created to act as a base. These are drivers that you will have already created on site.
IBM Drivers	Choose this option if you want to use an existing IBM driver to act as a base. These drivers have been supplied by IBM, and can be standard or SmartModel.
Choose an IBM Template	You will be asked to select which protocol you want to use, for example, CLI-based via Telnet/ SSH, or TL1. The protocols available to you will be listed.

Note: On the first occasion you create a new custom driver, the custom driver type will be greyed out. This is because you have not yet created a custom driver. In this instance, the IBM driver type is automatically selected.

4. The **Driver Details** window displays. Using the following table as a guide, enter the appropriate information requested. When the driver details have been specified, select **Next** to proceed.

   Note: Use the arrows in the centre of the screen to copy values across from the previously selected base driver. These values can then be modified to suit the new custom driver.

Table 5. Driver Details	
Selection	**Description**
Vendor/Type/Model/OS	These values have been copied from the base drivers selection.
Supported Models/Supported OSs	A list of the Models and OSs supported by the base driver chosen.

5. The **Default Driver Device Script** window displays.

   You can modify the Device Script to suit the requirements of the new custom driver. Or, you can load a new device script.

   Note: The device script needs to match the commands that would be entered manually into the device as well as the commands that handle a specific customer environment (for example, the banners). See Related concepts and Related tasks for information about Device Scripts and RADs.

   Select **Next** to proceed.
6. The **Default Driver RAD** window displays. You can modify the RAD, if necessary. Select **Next** to proceed.
7. The **Description** window displays. You should provide a description for the new custom driver. Select **Finish** to complete the new custom driver creation.

8. A summary screen will be displayed. This reports all details of the custom driver you have just created, including: UUID, Vendor, Type, Model, OS and display name.
9. When the custom driver is saved, it is written into the following files in the drivers section of the filesystem: `/opt/IBM/tivoli/netcool/ncm/drivers/locators/drivers/<newdriveruuid>.xml`, `/opt/IBM/tivoli/netcool/ncm/drivers/metadata/<newdriveruuid>/devicescript.txt` and `/opt/IBM/tivoli/netcool/ncm/drivers/metadata/<newdriveruuid>/rad.xml`. Your changes will be automatically applied to all presentation and worker servers.

Editing a custom driver

Once custom drivers have been created they can be edited to meet changing requirements. Edited custom drivers are automatically distributed to all presentation and worker servers.

The user must be a member of a group who has the "View System" activity. This enables the user to see the Systems Manager in the UI.

You should also understand how to work with Device Scripts and Resource Access Documents (RADs) before editing a custom driver. See Related concepts and Related tasks for information about Device Scripts and RADs.

Use this procedure to edit a custom driver.

1. Navigate to the Systems Manager, and select the Drivers component.
2. From the menu bar, select **Tools** > **Edit Driver**
3. The **Choose existing custom driver** window displays, allowing you to choose a driver to edit. Select the driver you want to edit from the list provided. The fields available in this window are: vendor, type, model, OS, UUID and version. Optionally, the filter drop downs can be used to narrow the number of rows visible in the table. Select **Next** to proceed.

   Note: Editable drivers are drivers which are not in production.
4. The **Driver Details** window displays. Use the arrows in the centre of the screen to copy values across from the selected driver. The existing values can be modified as required. When the modified driver details have been specified, select **Next** to proceed.
5. The **Default Driver Device Script** window displays.

   You can modify the Device Script to suit the changing requirements of the custom driver. Or, you can load a new device script.

   Note: The device script needs to match the commands that would be entered manually into the device as well as the commands that handle a specific customer environment (for example, the banners). See Related concepts and Related tasks for information about Device Scripts and RADs.

   Select **Next** to proceed.
6. The **Default Driver RAD** window displays. This can be modified if necessary, using the form or XML view. Select **Next** to proceed.
7. The **Description** window displays. You should provide a description for the edited custom driver. Select **Finish** to complete the modifications to the custom driver.
8. A summary screen will be displayed. This reports all details of the custom driver you have just edited, including: UUID, Vendor, Type, Model, OS, version, display name, and state. Your changes will be automatically applied to all presentation and worker servers.

Move a custom driver to production

Netcool Configuration Manager allows you to move a custom driver into production. A custom driver which is in production can be used as the basis for the creation of a new custom driver. Custom drivers in production provide more predictable behaviour, as they are non-editable.

The user must be a member of a group who has the "View System" activity. This enables the user to see the Systems Manager in the UI.

Use this procedure to move a custom driver to production.

1. Navigate to the Systems Manager, and select the Drivers component.
2. From the menu bar, select **Tools** > **Move Driver to Production**
3. The **Move driver to production** window displays. Select a row in the table to choose the custom driver to move into production. Optionally, the filter drop downs can be used to narrow the number of rows visible in the table. Select **Move to Production** to proceed.
4. A summary screen will be displayed. This reports all details of the custom driver you have just moved into production, including: UUID, Vendor, Type, Model, OS, display name and driver state.

   Note: A driver that has been moved to production cannot be edited, but it may be deleted.

Delete a custom driver

Netcool Configuration Manager allows you to delete either a single driver, or multiple drivers which can be deleted concurrently. A driver can only be deleted if it has an inactive status.

A driver can only be deleted if it has an inactive status.

Use this procedure to delete a custom driver.

1. Navigate to the Systems Manager, and select the Drivers component.
2. Highlight the driver(s) that you wish to delete, then right click and select **Delete Driver**.
3. The driver will be removed from the drivers table.

Importing a custom driver

Netcool Configuration Manager provides the ability to import custom drivers exported from other Netcool Configuration Manager systems. The imported driver will contain driver details, the device script and the RAD. Imported custom drivers are automatically distributed to all presentation and worker servers.

The user must be a member of a group who has the "View System" activity. This enables the user to see the Systems Manager in the UI.

Use this procedure to import a custom driver.

1. Navigate to the Systems Manager, and select the Drivers component.
2. From the menu bar, select **Tools** > **Import Driver**
3. The **Import driver ZIP file** window displays.
4. Navigate to the directory from which you wish to import the custom driver. Select the file, and then select **Open** to proceed.
5. A dialog will inform when a successful import has been made. Your changes will be automatically applied to all presentation and worker servers.

Exporting a custom driver

Netcool Configuration Manager provides the ability export a single custom driver to a user defined location in a zip file format. The exported driver will contain driver details, the device script and the RAD.

The user must be a member of a group who has the "View System" activity. This enables the user to see the Systems Manager in the UI.

Use this procedure to export a custom driver.

1. Navigate to the Systems Manager, and select the Drivers component.
2. From the menu bar, select **Tools** > **Export Driver**.
3. The **Export driver dialog** window displays. Select a row in the table to choose the custom driver to export. Optionally, the filter drop downs can be used to narrow the number of rows visible in the table. Select **Export** to proceed.

4. A **Save** window displays. Navigate to the directory where you wish to export the custom driver. Select **Save** to proceed.
5. A dialog will inform when a successful export has been saved.
6. The exported zip file consists of three files: `deviceScript.txt`, `driver.xml` and `rad.xml`.

Exporting a driver/server to CSV file

Netcool Configuration Manager provides the ability export drivers and servers to a user defined location in a CSV file format.

Use this procedure to export either drivers or servers to a CSV or txt format.

1. Navigate to the Systems Manager. Depending if you wish to export drivers or servers, choose the appropriate component.
2. Choose the items you wish to export.
3. A **Save** window displays. Navigate to the directory where you wish to export to, and choose the file type required. Select **Save** to proceed.
4. A dialog will inform when a successful export has been saved.

Set custom driver to active

Netcool Configuration Manager allows you to set drivers to an active or inactive state. This can be used to control the set of installed drivers that are applicable to the devices on your system. Multiple drivers can be activated concurrently. When a driver is set to active, it is loaded by devices, command sets and configurations, and it is eligible for optimality.

The driver status must be inactive, so that the menu option to **Set Driver Active** is available.

Use this procedure to activate a custom driver.

1. Navigate to the Systems Manager, and select the Drivers component.
2. Highlight the driver(s) that you wish to activate, then right click and select **Set Driver Active**.
3. The driver status will update to Active, and the driver details appear in green text.

Set custom driver to inactive

When a driver is set to inactive, it is no longer loaded by devices, command sets and configurations, and it is not eligible for optimality. Drivers that are "In Use" can be set to inactive. Inactive drivers appear as Incompatible in the Resource Browser search.

The driver status must be active, so that the menu option to **Set Driver Inactive** is available.

Note: Whilst drivers that are "In Use" can be set to inactive, an error message will be generated to ensure that you wish to continue. Setting an "In Use" driver to inactive affects command sets, command set groups, extractions and extraction groups, configurations, compliance definitions and extractions.

Use this procedure to set a custom driver to inactive.

1. Navigate to the Systems Manager, and select the Drivers component.
2. Highlight the driver(s) that you wish to set to inactive, then right click and select **Set Driver Inctive**.
3. The driver status will update to Inactive, and the driver details appear in red text.

Driver Reload

Netcool Configuration Manager provides the ability to dynamically load new drivers, so that drivers become available without the need to restart the server.

Newly created drivers will be detected and reloaded automatically. Drivers will also be reloaded when new custom drivers are created through the GUI or the command line. Users can trigger a driver reload manually using the **`Tools>Reload Drivers`** menu option. Typically this manual step will be taken after running the `SmartModelUpgrade` tool.

To enable dynamic reloading each JVM (worker or presentation) caches it's own copy of all driver jars and all third party dependencies. They are cached in a directory called `ncm/drivers/lib_legacy_cache_<server_name>`.

There are some instances where a restart will be required. If the jars stored in the cache change when a driver is installed, then a restart will be required. Also, if the driver interface jar is patched by the driver installer, a restart will be required. If a restart is required, the Driver Reload State column in the Server table will indicate this as "Restart Required". When the driver reload has been successful, the Driver Reload State column in the Server table will report this as "Reloaded".

Troubleshooting drivers

Use this section to view drivers troubleshooting information.

Custom driver may add leading 0x0a to password when communicating with devices

If Netcool Configuration Manager appends a leading 0x0A to the password, thereby causing connectivity issues, edit he deviceComInterface.txt as described in this workaround.

Workaround

Change `com.intelliden.icos.util.handlers.TelnetComHandler` to `com.intelliden.drivers.util.handlers.SocketComHandler` in the deviceComInterface.txt file. SocketComHandler is a raw connection and will ignore all the hand shake information and will just send and receive the data that Netcool Configuration Manager and the device are giving it.

Follow these steps to determine the location of deviceComInterface.txt used:

1. Run `./driverTools.sh -show-details <arg>`
   where <arg> can be UUID, VTMOS or network resource name.
   Example output:

```
[icosuser@klxv0803:/opt/IBM/tivoli/netcool/ncm/bin/utils]
08:08:37> ./driverTools.sh -show-details Riverbed_20131204
Loading driver information from database...
-------------------------------------------------------------
 UUID                  : IBM-fc43b18e-805b-4554-befc-40ffc5bf53ee
 Vendor                : Riverbed
 Type                  : wan
 Model                 : steelhead-xx50
 OS                    : 8.x
 Supported Models      : 550,1050,2050,6050,7050
 Supported Oss         : 8.*
 Display Name          : riverbed_wan_steelhead-xx50_8.x_v20131203.083318
 Version               : 20131203.83318
 Description           :
-----------------------------------------------------
 Type                  : SmartModel
 Driver Class          : com.ibm.tivoli.ncm.drivers.riverbed.RiverbedDriver
 Jar File              : Jar-b63a722b-34ba-482a-9eaa-3bebda2a1e09.jar
 Metadata              : Metadata-24902fdb-01d7-4a61-afa2-1af6f054795e
 Metadata              : Metadata-03e0700f-e390-4f42-ad53-906af4f81f3e
 Metadata              : Metadata-7354e070-d920-444b-997f-55c7361ca768
 Metadata              : Metadata-b5f684ab-ee2e-4cf7-a9e8-bf91274e628e
 Master Schema Version : 201311290917
 Schema Object Version : BL_ITNCM_CONTENT_19_20131129_0854
 Definition Doc. Att.  : "8.x,  1311282011"
-------------------------------------------------------------
```

2. Check the list of metadata folders for the deviceComInterface.txt location.
   Example deviceComInterface.txt location: `/opt/IBM/tivoli/netcool/ncm/drivers/metadata/Metadata-03e0700f-e390-4f42-ad53-906af4f81f3e`
3. Change the deviceComInterface.txt file as follows:
   `telnet:com.intelliden.drivers.util.handlers.SocketComHandler:`

```
ssh:com.intelliden.icos.util.handlers.SSHComHandler:
```

4. Restart the servers.

Related tasks

Driver lifecycle

You can use the Systems Manager to access, manage, and control multiple servers and drivers. Your window displays only those system resources for which you have authorization for display or control.

Related information

Custom Drivers

Netcool Configuration Manager drivers encapsulate all vendor-specific operations and provide a layer of abstraction that allows Netcool Configuration Manager to remain vendor agnostic. For example, drivers enable Netcool Configuration Manager to communicate with the different devices used within your network, retrieve configuration, make configuration changes and so on.

Chapter 6. Scripts and utilities

Use the supplied Netcool Configuration Manager scripts and utilities to perform system administration tasks.

Administering Netcool Configuration Manager scripts

The ITNCM - Base application provides scripts to use in system administration tasks. These scripts provide the user with the ability to undertake a broad range of roles. The scripts are configurable, and therefore provide the user with flexibility.

Specifically, the ITNCM - Base application provides the following categories of scripts:

- System scripts — These scripts perform such system administration tasks as restarting ITNCM - Base, collecting and storing in an archive data that resides in the ITNCM - Base install directory, and so forth.
- IDT scripts — These scripts perform system administration tasks related to IDT.
- Logging scripts — These scripts control logging activities.

System scripts

The system scripts are located in the following directory: `/opt/IBM/tivoli/netcool/ncm/bin/utils`.

Summary descriptions of system scripts

The following table provides descriptions of each system script:

Script name	Description
`createAutoStart.sh`	In the event that the server is rebooted, this script will automatically restart ITNCM - Base.
`dataCollector.sh`	Requests base ITNCM - Base install directory, all relevant data located here is collected and stored in an archive.
`driverTools.sh`	Provides the user with the ability to manage device drivers using the command line.
`itncm.sh`	Starts the ITNCM - Base server.
`logcleaner.sh`	Used to clear down the logs (For Fix Pack 11 and earlier versions). The logcleaner.sh script file is not functional for Fix Pack 12 and above.
`logcleaner.cnf`	Used to clear down the logs (For Fix Pack 12 and later versions).
`portUsage.sh`	Shows the ports currently in use by the product.
`truncateDrivers.sh`	Removes content from the drivers table, but does not remove the schema.

Managing device drivers (driverTools.sh)

Use the `driverTools.sh` system script to manage device drivers from the command line.

Syntax

```
driverTools.sh -show-details args -consistency-check -create-custom-cli
-create-custom-tl1 -help
```

Parameters

-show-details *args*
Shows detailed information about a specific driver:

-consistency-check
Consistency check the drivers on the filesystem against the database.

-create-custom-cli
Creates a new custom Driver.

-create-custom-tl1
Creates a new custom tl1 Driver.

-help
Displays an overview of available options.

Samples

The following sample shows an INFO statement that can be ignored: `Invalid classpath entry`.

```
./driverTools.sh -consistency-check
Starting driver consistency check ...
2020.04.24 16:28:38 GMT+00:00 INFO    com.intelliden.admin.ServerRegistrar :: Server type: NONE
2020.04.24 16:28:38 GMT+00:00 INFO    com.intelliden.admin.ServerRegistrar :: Server name: null
2020.04.24 16:28:41 GMT+00:00 INFO    com.intelliden.drivers.DriverClassLoaderFactory ::
Invalid classpath entry. Files [commons-1.4.1-net.jar, icos-legacy.jar, jsch-0.1.37.jar] not
found in /opt/IBM/tivoli/netcool/ncm/drivers/lib_legacy_cache/
2020.04.24 16:28:41 GMT+00:00 INFO    com.intelliden.drivers.DriverLocator :: Driver
consistency check started ...
2020.04.24 16:28:45 GMT+00:00 INFO    com.intelliden.drivers.DriverConsistencyCheck :: No new
drivers to write to the database
2020.04.24 16:28:45 GMT+00:00 INFO    com.intelliden.drivers.DriverLocator :: Post driver
consistency check new drivers was empty
2020.04.24 16:28:45 GMT+00:00 INFO    com.intelliden.drivers.DriverLocator :: Standard driver
is IBM-ffea5723-205a-4cb4-b9e9-ac0797a17a4e
2020.04.24 16:28:45 GMT+00:00 INFO    com.intelliden.drivers.DriverLocator :: Driver
consistency check complete (3545 ms)
```

The following sample shows information by IBM uuid from the UoW Log:

```
driverTools.sh -show-details IBM-ABC1234-ABCD-1234-1a2b-a12sbvn472m9
```

The following sample shows information by device name in the UI:

```
driverTools.sh -show-details 10.1.2.3
```

The following sample shows information by by VTMOS: driverTools.sh

```
-show-details Cisco Router 2600 12.3
```

See also

“Modifying logging” on page 94

Related reference
Custom driver capabilities

Custom drivers (and standard drivers) differ in functionality from SmartModel drivers, which is described here.

Performing operations on the Netcool Configuration Manager server (itncm.sh)

Use the `itncm.sh` system script to perform start, stop, and restart the Netcool Configuration Manager server. You can also use `itncm.sh` to get current status information on the Netcool Configuration Manager server.

Syntax

```
itncm.sh {start|stop|restart|status}
```

Description

There are a number of different tasks that can be performed on the Netcool Configuration Manager server. For example, the server can be started, stopped, restarted or the current status checked. The following steps describe how to execute the `itncm.sh` system script:

1. Access the directory containing the installer. The default location is `/opt/IBM/tivoli/netcool/ncm/bin`.
2. To use the example of starting the Netcool Configuration Manager server, execute the start server command:

   ```
   $./itncm.sh start
   ```

When executing the Netcool Configuration Manager stop server command, the superuser username and password are prompted for as follows:

```
Stopping GUI Server
Realm/Cell Name: <default>
Username: Superuser
Password: ****
```

Parameters

Fix Pack 2

start
Start all components.

start-worker
Start the worker component.

start-comp
Start the compliance component if installed.

start-pres
Start the presentation component if installed.

stop
Stop all components.

stop-worker
Stop the worker component.

stop-comp
Stop the compliance component if installed.

stop-pres
Stop the presentation component if installed.

restart
Stop and start all components.

restart-worker
Stop and start the worker component.

restart-comp
Stop and start the compliance component if installed.

restart-pres
Stop and start the presentation component if installed.

status
Show the status of the server.

IDT scripts

The IDT scripting utilities may be found using the path: `/opt/IBM/tivoli/netcool/ncm/bin/utils/idt`.

The following table provides a descriptions of the IDT script.

Table 6. IDT scripts

Script name	Description
`changeIDTDaemon.sh`	This script is used to convert 'presentation server' SSH Daemon to 'mainserver' or 'server' mode. **Usage** `changeIDTDaemon.sh <mode>` where mode can be mainserver or server. Use 'mainserver' if you want one presentation server to be the dedicated route for all IDT sessions. Use 'server' if you want IDT sessions to go through the presentation server the user is connected to.

Netcool Configuration Manager - Compliance scripts

The Netcool Configuration Manager - Compliance application provides scripts used in system administration tasks.

Purpose

The Netcool Configuration Manager - Compliance application scripts provide the user with the ability to undertake a broad range of roles. The scripts are configurable, and therefore provide the user with flexibility. The `utils` directory can be found using the following path:

`/opt/IBM/tivoli/netcool/ncm/compliance/bin/utils`

Note: Restart the Netcool Configuration Manager - Compliance srver if the system date or time is changed, as this may affect the execution of scheduled processes.

Parameters

The following table provides the information about the compliance scripts that reside in the `utils` directory. Specifically, the table provides the following:

- Script name
- Description of the script

Script name	Description
createAutoStart.sh	Automatically restarts the Netcool Configuration Manager - Compliance application in the event that the server is rebooted.
dbExport.sh	Extracts entire tables from the database.
dbImport.sh	Imports XML dataset into the database specified in the Database Properties file.
deviceSchemaLoader.sh	Loads device content into Netcool Configuration Manager - Compliance.
houseKeeping.sh	Deletes old, unwanted process results
intellidenRmUser.sh	Changes the usernames and passwords of Netcool Configuration Manager - Compliance system users. For more information, see *Change Compliance user names and password using the intellidenRmUser.sh script* and *Additional group permissions* in the *IBM Tivoli Netcool Configuration Manager Administration Guide*.
logcleaner.sh	Used to perform house keeping operations on logs (For Fix Pack 11 and earlier versions). Depending on the parameters passed, logs may be compressed or deleted.
logcleaner.cnf	Used to perform house keeping operations on logs (For Fix Pack 12 and later versions). Depending on the parameters passed, logs may be compressed or deleted.
policyExport.sh	Extracts all Policy data from the database.
policyImport.sh	Imports all previously exported Policy data back into the database.
RefreshSecurityTables.sh	Synchronizes security privileges from Netcool Configuration Manager - Base into Netcool Configuration Manager - Compliance.
updateModelledDefinitions.sh	Updates Modelled Definitions that were defined using the incorrect VTMOS combination.
upgradeDB.sh	Upgrades database schema.

Administering logs

The logging scripts perform logging operations on Netcool Configuration Manager.

Summary descriptions of IDT scripts

The following table provides descriptions of each logging script:

Script name	Description
`loggerAdmin.sh`	Controls logging operations for Netcool Configuration Manager Core and User Interface.
`logcleaner.sh`	Performs housekeeping operations on the logs (For Fix Pack 11 and earlier versions). The logcleaner.sh script file is not functional for Fix Pack 12 and above.
`logcleaner.cnf`	Performs housekeeping operations on the logs (For Fix Pack 12 and later versions).

About logging

To build the default logging configuration, a server gathers data from a number of locations, which are described here..

Tip: To turn on logging for Netcool Configuration Manager Drivers do the following:

```
./bin/loggerAdmin.sh -a -c legacy.com.intelliden -c com.intelliden -l TRACE
```

This will outputs to `ncm/logs/worker.log`

Default logging data is gathered in the following locations:

logger.properties server name section, and logger.properties server type section

'logger'-prefixed entries are entries that provide log4j configuration lines.

Entries that start with 'type' will specify commands for specific types of servers (e.g. WORKER, GUI_SERVER, etc).

These are mostly unused, unless the presentation server or compliance server are configured to use the **`mainLogPort`** on a network logging append.

logger.properties

Note: This file has been updated for fix pack 2. Only the **`mainLogPort`** and **`listenPort`** configuration options are still in use.

mainLogPort

Specifies the port that the logging server will listen on for `Log4J` log messages. If not specified, the listener is not enabled.

mainLogPort=8103

listenPort

Specifies the port that the logging server will listen on for `LoggerAdmin.sh` commands.

Sane defaults

```
listenPort=8102
log4j.rootLogger=INFO, CONSOLE
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.targe=System.out;
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=%d{HH:mm:ss} %-5p
%X{serverType} %X{serverName} %-25c{1} :: %m%n
```

Modifying logging

Fix Pack 2 Use the `loggerAdmin.sh` script to modify logging files. Logging for the Netcool Configuration Manager Core and User Interface is controlled by the `loggerAdmin.sh` script, which is located

in /opt/IBM/tivoli/netcool/ncm/bin. To avoid performance issues, you must switch back to normal logging levels after using debug levels.

Syntax

```
loggerAdmin.sh [-a] [-c <arg>] [-h] [-i <arg>] [-l <arg>] [-o <arg>] [-p <arg>]
 [-q] [-r] [-s
    <arg>]
```

Parameters

-a,--set
Set value

-c,--component *<arg>*
The specific component logging to change. If not specified, then the default is changed.

-h
Help

-i,--initialize *<arg>*
The initialize command reads a local log4j configuration file, and then sends it to the remote server to configure the logging.

Reset to configuration using the specified file. Must be followed by one file name.

-l,--level *<arg>*
Set logging level. No effect for get command.

Must be one of 'TRACE', 'DEBUG', 'INFO', 'WARN', 'ERROR', or 'FATAL'.

If not specified, resets to the current default value.

-o,--output *<arg>*
Output file.

If not specified, writes to standard out.

-p,--port *<arg>*
The port to be used for the specified server.

Not applicable if 'server' is not specified.

-q,--query
Query values.

If 'server' is not specified, then either a list of servers will be generated, or a list of configuration options for the server will be generated.

-r,--reset
The reset command resets the logging configuration to the initial startup settings.

-s,--server *<arg>*
The specific server or servers change.

If not specified, then all servers are changed.

Note: A command must contain one, and only one, 'query', 'set', 'reset', or 'initialize' operation.

Examples

The following example sets the server to DEBUG logging

```
loggerAdmin.sh -a -l DEBUG -s url.for.server
```

The following example sets the logging level for all `com.intellliden.data` access classes, and all classes in the sub packages to DEBUG

```
loggerAdmin.sh -a -l DEBUG -c com.intelliden.dataaccess -s url.for.server
```

The following example configures the logging server based on the contents of the `myConfig.file`

```
loggerAdmin.sh -s url.for.server -i myConfig.file
```

Warning: You must switch back to normal logging levels after using debug levels, to avoid performance issues.

Presentation server logging

The presentation server logs all information to the WebSphere Application Server (WAS) logging system.

This allows the WAS console to adjust the logging and logging output of the presentation server(s) on demand, and while the systems executes.

Compliance server logging

The compliance server uses Log4J, and the logging configuration is specified in a standard Log4J configuration file stored in the following location: *<InstallDir>*`/compliance/config/properties/logging/log4j.properties`

Note: From FP16 onwards, the configuration file is stored in the following location: `<InstallDir>/compliance/config/properties/logging/log4j2.properties`

To change the logging levels and other settings, you modify the file, then stop and restart the server.

Worker server logging

The worker server uses Log4J and has a default configuration file stored in the following location: *<InstallDir>*`/config/properties/logging/log4j.properties`.

Note: From FP16 onwards, the configuration file is stored in the following location: `<InstallDir>/config/properties/logging/log4j2.properties`

After the log file reaches a certain size, a new log file is created. To configure the maximum size of a log file, set the `log4j.appender.FILE.MaxFileSize` property, as shown in the following example:

```
# Set the maximum file size before rollover
log4j.appender.FILE.MaxFileSize=5MB
```

Note: From FP16 onwards, change the below parameter.

```
appender.rolling.policies.size.size=3MB
```

To configure how many log files are created before the first file is overwritten, set the `log4j.appender.FILE.MaxBackupIndex` property, as shown in the following example:

```
# Set the backup index
log4j.appender.FILE.MaxBackupIndex=10
```

Note: From FP16 onwards, change the below parameter.

```
appender.rolling.strategy.max=10
```

Important: The worker has an embedded logging server that allows dynamic reconfiguring of the logging without stopping and starting the server. However, the dynamic changes will be reset to the defaults when the server is restarted.

Driver logging

Driver logging occurs in the logs of the system that made the call to the driver. For example, if a worker server calls the driver code, then the driver logging will be in the Worker server logs.

To increase logging to debug levels for troubleshooting and log collection by the support teams, change to the `/bin` directory within the installation directory and run the following statements as the same user that installed Netcool Configuration Manager:

```
./loggerAdmin.sh -a -l TRACE -c com.ibm.tivoli.ncm.drivers.utils
./loggerAdmin.sh -a -l TRACE -c com.intelliden.icos.util.handlers
./loggerAdmin.sh -a -l TRACE -c com.intelliden.icos.irm.PhysicalDevice
./loggerAdmin.sh -a -l TRACE -c com.intelliden.icos.util.socket
./loggerAdmin.sh -a -l TRACE -c com.intelliden.icos.util.socketSSH2Wrapper
./loggerAdmin.sh -a -l TRACE -c com.intelliden.icos.util.socket.ComWrapper
./loggerAdmin.sh -a -l TRACE -c com.ibm.tivoli.ncm.drivers.comms.socket.SSH2IO
./loggerAdmin.sh -a -l TRACE -c com.intelliden.icos.util.socket.TelnetIO
./loggerAdmin.sh -a -l TRACE -c com.intelliden.icos.util.DeviceComInterfaceFactory
./loggerAdmin.sh -a -l TRACE -c com.intelliden.icos.irm.PhysicalDeviceFactory
./loggerAdmin.sh -a -l TRACE -c legacy.com.intelliden.icos.util.handlers
./loggerAdmin.sh -a -l TRACE -c legacy.com.intelliden.icos.util.socket
./loggerAdmin.sh -a -l TRACE -c com.intelliden.drivers.omnidriver
```

To reset the logging levels when you no longer need them to be at debug level, run the following commands:

```
./loggerAdmin.sh -a -l ERROR -c com.ibm.tivoli.ncm.drivers.utils
./loggerAdmin.sh -a -l ERROR -c com.intelliden.icos.util.handlers
./loggerAdmin.sh -a -l ERROR -c com.intelliden.icos.irm.PhysicalDevice
./loggerAdmin.sh -a -l ERROR -c com.intelliden.icos.util.socket
./loggerAdmin.sh -a -l ERROR -c com.intelliden.icos.util.socketSSH2Wrapper
./loggerAdmin.sh -a -l ERROR -c com.intelliden.icos.util.socket.ComWrapper
./loggerAdmin.sh -a -l ERROR -c com.ibm.tivoli.ncm.drivers.comms.socket.SSH2IO
./loggerAdmin.sh -a -l ERROR -c com.intelliden.icos.util.socket.TelnetIO
./loggerAdmin.sh -a -l ERROR -c com.intelliden.icos.util.DeviceComInterfaceFactory
./loggerAdmin.sh -a -l ERROR -c com.intelliden.icos.irm.PhysicalDeviceFactory
./loggerAdmin.sh -a -l ERROR -c legacy.com.intelliden.icos.util.handlers
./loggerAdmin.sh -a -l ERROR -c legacy.com.intelliden.icos.util.socket
./loggerAdmin.sh -a -l ERROR -c com.intelliden.drivers.omnidriver
```

Note: This does not include any output that the driver sends to `System.out` or `System.err`

- "Managing device drivers (driverTools.sh)" on page 90
- "Performing housekeeping on log files (Fix Pack 11 and earlier)" on page 98

Configure user and group audit logging

You can enable and disable logging of user and group actions.

User and group interactions with Netcool Configuration Manager can be logged to file. Logging is on by default. The default file is `/opt/IBM/tivoli/netcool/ncm/logs/audit.log`.

The log file contains the following types of information:

- Successful and unsuccessful user logins to Netcool Configuration Manager through the Java API, NSN, Netcool Configuration Manager presentation server, configuration GUI, or compliance GUI.
- Details of the creation, inactivation, reactivation, and modification of Netcool Configuration Manager user definitions.
- Details of the creation, deletion, and modification of Netcool Configuration Manager group definitions.

Tip:

When a user authenticates using the Netcool Configuration Manager Java API, the authentication is logged to the `audit.log` file. Subsequent connections by that user might not be logged, because the user has already authenticated and the user session contains the authentication token.

Whether the user needs to authenticate again depends on the settings in the Websphere Application Server console for Authentication cache settings timeout and LTPA Token time out.

For more information about these settings, refer to the information about *authentication cache timeout* and *configuring the Lightweight Third Party Authentication mechanism* in the Websphere Application Server Knowledge Center at: https://www.ibm.com/support/knowledgecenter/en/SSEQTP/mapfiles/product_welcome_was.html

To configure user and group audit logging, complete the following steps:

1. Edit the following file: `/opt/IBM/tivoli/netcool/ncm/config/properties/logging/log4j.properties`.
2. Change the value of the `log4j.appender.AUDITFILE.Threshold` property to `OFF` to disable the logging, or `ON` to enable it.
3. Restart the presentation server.

Below is an example of the type of information logged to file for the creation and removal of a user `Tom123`.

```
2019.10.03 11:59:58 GMT+00:00 WebContainer : 5
INFO    com.intelliden.icos.iweb.servlet.LoginFilter ::
User administrator logged into DASH server
2019.10.03 12:00:30 GMT+00:00 WebContainer : 3
INFO    com.intelliden.datawrapper.accounts.handlers.AbstractHandler ::
User Tom123 was created by user administrator.
User Tom123 has group/groups 'observer'  assigned
2019.10.03 12:00:36 GMT+00:00 WebContainer : 1
INFO    com.intelliden.datawrapper.accounts.handlers.AbstractHandler ::
User Tom123 was inactivated by user administrator
```

Performing housekeeping on log files (Fix Pack 11 and earlier)

Use the `logcleaner.sh` script to perform housekeeping on log files for Fix Pack 11 and earlier versions.

Syntax

```
./logcleaner.sh compress-after [+]days [delete after [+]days]
```

Parameters

`compress-after` *days*

Compress the contents of the log if it is *days* old.

`compress-after` + *days*

Compress the contents of the log if it is more than *days* old.

`delete-after` *days*

Delete the contents of the log if it is *days* old.

`delete-after` +*days*

Delete the contents of the log if it is more than *days* old.

Description

The `logcleaner.sh` script executes on logs to remove old, unwanted logging information. This utility is used to free up storage space.

The `logcleaner.sh` script resides in `/opt/IBM/tivoli/netcool/ncm/bin/utils`. This script can be run as a cron job.

This should only be executed on the main presentation server.

Note: The logcleaner.sh script file is not functional for Fix Pack 12 and above. However, you can use `logcleaner.cnf` file to remove old and unwanted logs.

See also

- “Managing device drivers (driverTools.sh)” on page 90
- “Modifying logging” on page 94

Performing housekeeping on log files (Fix Pack 12 and later)

Use the `logcleaner.cnf` script file to perform housekeeping on log files for Fix Pack 12 and later versions. The logcleaner.sh file is obsolete for fix pack 12 and later versions.

Syntax

You can use the `logcleaner.cnf` script file to remove old, unwanted logging information to free up storage space. To run the `logcleaner.cnf` script, logrotate must be installed. For detailed information about logrotate, refer to the logrotate documentation.

The logcleaner.cnf script resides in `/opt/IBM/tivoli/netcool/ncm/bin/utils`.

This script can be run as a cron job. For detailed information about cron job, refer to the cron job documentation.

You can change the configuration parameters based on the requirement. For example: Rotate and Size.

Following is the example script for `logcleaner.cnf` script file.

```
INSTALL_DIR/logs/Server.err {
                daily
                rotate 10
                dateext dateformat-%Y-%m-%d-%H-%s
        compress
                missingok
        notifempty
                size 10M
}
```

Following logs can be removed by using logcleaner.cnf script file:

- CMServer.out
- CMServer.err
- Server.err
- Server.out
- IDT.err
- IDT.out

Note: This should only be executed on both worker and compliance servers. The `logcleaner.cnf` script file should not be moved from its location (otherwise the `archive()` function will create different results from the cron). If you have modified the `logcleaner.cnf` file, it must be preserved before you upgrade to the newer or subsequent versions. The `logcleaner.cnf` script file should be preserved manually.

See also

- “Modifying logging” on page 94

Viewing the compliance event log

The ITNCM-Compliance event log contains general information about the operations that occur within the application. View the event log by accessing the User Audit Trail from the User Interface.

To view the event log, follow these steps.

From the User Interface, select Admin | User Audit Trail. The event log is displayed. The following table describes each of the fields in the event log.

Option	Description
Screen item	Description
Time	Specifies the time in which the event took place. All times are server side.
Description	Provides a description of the event.
User	Specifies the username of the person who triggered the event.

The following shows examples of events in the event log:

```
Time                                    Description                                    User

07-Jul-2011                          User admin has logged in            admin
07-Jul-2011                          User admin has logged out        admin
```

Administering Netcool Configuration Manager utilities

Use the supplied Netcool Configuration Manager application utilities to perform a broad range of system administration tasks, including changing super user and database passwords and encrypting passwords.

The icosadmin utility

The `icosadmin` utility performs such tasks as changing super user and database passwords.

The `icosadmin` utility uses the following syntax:

`./icosadmin` *`admin_task admin_task_options`*

Where:

- `./icosadmin` — Invokes the `icosadmin` utility.
- *admin_task* — Specifies the keyword associated with a system administration task. The following table maps the keyword to its associated task:

icosadmin Keyword	Description
`ChangeSuPassword`	Changes the ITNCM - Base super user password.
`ChangeDbPassword`	Changes the Oracle or DB2 database password.
`LoggerAdmin`	Modifies logging-related files.

- *admin_task_options* — Specifies the options associated with the keyword specified in *admin_task*.

Note: ITNCM - Base should be restarted after the execution of the `icosadmin` utility with any of these keywords.

Changing the super user password (ChangeSuPassword)

To change the ITNCM - Base application super user password, specify the `ChangeSuPassword` keyword when executing the `icosadmin` utility.

Syntax

```
./icosadmin ChangeSuPassword ITNCM username Current Superuser password
ITNCM Server ITNCM Port New Superuser password
```

Parameters

ChangeSuPassword
Specifies the keyword that instructs the `icosadmin` utility to change the ITNCM - Base super user password.

ITNCM username
Specifies the name of the ITNCM - Base user whose associated super password is to be changed.

Current Superuser password
Specifies the current super user password that is to be changed.

ITNCM Server
Specifies the name of the server on which ITNCM - Base is running.

ITNCM Port
Specifies the port number.

New Superuser password
Specifies the new super user password to be associated with the user specified in *ITNCM username*.

Changing the database password (ChangeDbPassword)

To change the password on the database, specify the `ChangeDbPassword` keyword when executing the `icosadmin` utility. The `icosadmin` utility, when specified with the `ChangeDbPassword` keyword, changes the database password only when ITNCM - Base tries to log into the database. This procedure works for both Oracle and DB2 databases.

Syntax

```
./icosadmin ChangeDbPassword -u Database username -p Current database password
 -n New database password
```

Parameters

ChangeDbPassword
Specifies the keyword that instructs the `icosadmin` utility to change the database password.

Database username
Specifies the name of the database user whose associated password is to be changed.

Current database password
Specifies the current database password that is to be changed.

New database password
Specifies the new database password to be associated with the user specified in *Database username*.

Modifying logging-related files (LoggerAdmin)

To modify logging-related files, specify the `LoggerAdmin` keyword when executing the `icosadmin` utility.

Syntax

```
./icosadmin LoggerAdmin log-level [port]
refresh off classname on classname
maxline x maxsize x status
```

Parameters

LoggerAdmin
Specifies the keyword that instructs the `icosadmin` utility to modify logging-related files.

log-level
Specifies one of the following log levels. The log levels are provided in order of increasing detail.

- FATAL
- ERROR
- WARN
- INFO
- DEBUG

Note: ITNCM - Base does not need to be restarted for these changes to take effect.

port
Specifies the port number. This is an optional parameter

The auto-discovery utility

The `autodiscoveryUtil` utility is a Unix shell script that can restore a previously installed version of autodiscovery, display the current version of autodiscovery, and perform an update to the latest version of autodiscovery. It logs activities and displays results in the console.

Syntax

```
./autodiscoveryUtil.sh itncm_install_dir [ -r file | -u | -v]
```

Parameters

`itncm_install_dir`
Specify the home directory of the Netcool Configuration Manager installation.

`-r file`
Specify the restore option and full location of the autodiscovery zip file containing the previous version.

`-u`
Specify the update option to update the current autodiscovery to the version provided with the installer.

`-v`
Specify the version option to display the date and version of the installed autodiscovery file.

The icosutil utility

The `icosutil` utility performs such tasks as encrypting passwords and importing quantities of network resources into the database.

The `icosutil` utility uses the following syntax:

`./icosutil` *`admin_task admin_task_options`*

Where:

- `./icosutil` — Invokes the `icosutil` utility.
- *`admin_task`* — Specifies the keyword associated with a system administration task. An optional `-help` parameter can be appended after the `admin_task` to view the task's usage information.
- *`admin_task_options`* — Specifies the options associated with the keyword specified in *admin_task*.

The following table describes the function of each keyword.

icosutil keyword	Description
`Archive`	Archives the UOW.

icosutil keyword	Description
ArchiveDelete	Clears the archive.
BulkLoader	Imports quantities of network resources into the database.
CalculateChecksum	Used to calculate the check sum of a file and create a file with that value in it (only to be used when instructed to do so by IBM L2 Support).
ConfigArchive	Archives and deletes versioned configuration information from the database.
ConfigRestore	Reverses the ConfigArchive process by restoring a versioned configuration from the specified archive file.
CmdSetMigration	Used to update Command Sets whose driver is now out of date.
Encrypt	Encrypts the specified password.
IDTArchive	Archives IDT session logs from ITNCM - Base network resources.
Restore	Reverses the archive process by restoring UOW and UOW logs from the specified archive.
ReevaluateGroups	Determines if the addition or deletion of drivers has an impact on the coverage of the command set groups and updates them accordingly.
Fix Pack 10 ResourceUtility	Loads a given resource.
TaskSleepResume	Used to test drivers that support sleep and resume functionality (only to be used when instructed to do so by IBM L2 Support).
VerifyPermissions	Used to check all account permissions are correct for each group (only to be used when instructed to do so by IBM L2 Support).
WorkAnalyser	This utility is used to analyze and produce reports on failed UoWs for the support team.
WorkHousekeeping	Deletes completed UOWs from the database.

The options for each keyword are given in individual topics in this section, or, for keywords that relate to housekeeping, in the “Housekeeping” on page 115 section.

Loading a resource (ResourceUtility)

To populate the database with generalized resources created by users, specify the `ResourceUtility` keyword when executing the `icosutil` utility.

Syntax

```
./icosutil ResourceUtility -l username -p password -port port -f resource_file_path -r realm -n
resource_name -vtmos VTMOS -c resource_type
```

Parameters

`username`
Specifies a valid ITNCM - Base username. This user must belong to a group with add rights to the realm in which you are adding the resource, as well as add rights to resources in that realm.

password
Specifies the password associated with the ITNCM - Base username you are using. This must be clear text.

port
The port to connect on. By default, this is 16310.

resource_file_path
The fully qualified path to the resource file.

realm
The realm in which to add the resource.

resource_name
Specify the resource name. This will be the name that is displayed in the drop-down lists within the user interface.

VTMOS
Specify the VTMOS that this resource controls. Use the following format:

`Vendor/Type/Model/OS`

No validation is performed, so be sure you enter the VTMOS correctly.

resource_type
Specify the type of resource to add. Can be one of the following:

- SecuritySet
- Realm
- CommandSet
- NativeCommandSet
- Authentication
- FileTransfer
- ResourceAccDoc
- OSRegistry
- OSUpdate
- SearchSet
- Shortcut

Sample

The following example loads a RAD file called `device-R7a.xml`:

```
./icosutil ResourceUtility -l administrator -p administrator -port 16310 -f /home/icosuser/
Desktop/device-R7a.xml -r ITNCM/Test -n deviceR7a -vtmos Cisco/Router/asr1k/3.x -c ResourceAccDoc
```

Encrypting a password (Encrypt)

To encrypt a specified ITNCM - Base password, specify the `Encrypt` keyword when executing the `icosutil` utility.

Syntax

```
./icosutil Encrypt -u password
```

Parameters

Encrypt
Specifies the keyword that instructs the `icosutil` utility to encrypt the password specified in the *password* parameter.

-u
Specifies a required option when using the `Encrypt` keyword.

password
Specifies the password to be encrypted.

Sample

The following example specifies a sample password called `password`:

```
./icosutil Encrypt -u password
```

The `icosutil` utility would return the following:

```
Encrypted string is f4b37ba1e629
```

Command Set Migration (CmdSetMigration)

To update Command Sets after you install new drivers, use the `CmdSetMigration` utility. You execute the `CmdSetMigration` utility by specifying the `CmdSetMigration` keyword when executing the `icosutil` utility. This keyword executes the `CmdSetMigration` utility with the options specified on the command line.

Syntax

```
./icosutil CmdSetMigration -f /opt/IBM/tivoli/netcool/ncm/config/properties/
cmdSetMigration.properties
```

Description

The `CmdSetMigration` utility evaluates Command Sets whose driver or drivers are updated, based on criteria specified in the `cmdSetMigration.properties` file. By default, the `cmdSetMigration.properties` file resides in the `/opt/IBM/tivoli/netcool/ncm/config/properties` directory.

The Command Set Migration tool should be used after the Drivers are updated. The tool checks for Command Sets created with a driver that has since been updated, and verifies that the Command Set is still valid to use with the latest driver schema. It validates the Command Set XML against the latest schema. If the Command Set XML is no longer valid then the problematic nodes will be reported on the command line, and that Command Set will be marked as Incompatible in the GUI. Command Sets marked this way will need to be recreated. If the Command Set XML is still valid then the tool migrates the Command Set to use the latest version of the driver/schema when being opened or edited in the future.

The `CmdSetMigration` utility can update Command Sets based on the following criteria:

- `realm`
- `subrealmsFlag`
- `realmCmdSetnameFilter`
- `realmVendorFilter`
- `realmTypeFilter`
- `realmModelFilter`
- `realmOsFilter`
- `reportOnlyFlag`
- `performOldSchemaComparisons`

Parameters

CmdSetMigration
Specifies the keyword that instructs the `icosutil` utility to execute the `CmdSetMigration` utility. The `CmdSetMigration` utility proceeds to update Command Sets based on criteria specified in the `cmdSetMigration.properties` file.

-f
Specifies a file option when using the `CmdSetMigration` keyword.

cmdSetMigration.properties
Specifies the `cmdSetMigration.properties` file that defines the criteria that the `CmdSetMigration` utility uses to update Command Sets whose driver is out of date.

`cmdSetMigration.properties` can be updated to run against Command Sets in a particular realm, those with a particular VTMOS, or a Command Set with a specific name. The `performOldSchemaComparisons` flag in this file can be used to check for nodes whose descriptions have changed. If the user does not want this scenario to fail the validation of the Command Set then leave this flag set to false.

Below are two examples of errors you may encounter when running the tool:

1.

```
ERROR: Xpath "configuration||interfaces||interface_20sp-
||unit||service-domain||inside" not found in schema
```

This means that this node no longer exists in the new schema. The node may have been renamed, moved, or removed completely from the new schema.

2.

```
ERROR: Invalid value "64000" found for element at xpath "configuration||
logging||buffered||ARG.002" with description "0-7 -  Logging severity level"
```

This means the value in this node is no longer valid in the new schema, new validation may have been added or the existing validation may have been changed

Some users may find some of their Command Sets are very large and cause the tool to run out of memory, if this is the case then the memory can be increased for the tool by editing: `/opt/IBM/tivoli/netcool/ncm/bin/icosutil` and changing `MAXHEAP` to a larger value, e.g.: Change `MAXHEAP=-Xmx1024m` to `MAXHEAP=-Xmx1750m`.

Netcool Configuration Manager - Compliance utilities

Use the supplied Netcool Configuration Manager - Compliance application database utilities to import and export policies to and from the Netcool Configuration Manager - Compliance database. Use the supplied Netcool Configuration Manager - Compliance application policy utilities to import and export policies to and from the Netcool Configuration Manager - Compliance database.

dbExport

The `dbExport` utility extracts all policies from the database.

Purpose

The `dbExport` utility extracts all policies from the database, and places the extracted data in flat XML files. The utility creates only one XML file for each table extracted from the database.

Syntax

To invoke the `dbExport` utility, use the following syntax:

`./dbExport`

Tables exported

By default, the dbExport utility extracts the following tables from the database:

- POLICY
- POLICYTOALARMACTION
- POLICYTOEMAILACTION
- POLICYTORULE
- RULE
- RULEBLOB
- RULETODEFINITION
- RULETOCOMMANDSETACTION
- DEFINITION
- EVALUATION
- COMMANDSETACTION
- EMAILACTION
- ALARMACTION
- GLOBALPARAMETER
- EXTRACTION
- PARAMETER-GROUP
- PARAMETERGROUPVALUE

Output

The dbExport utility creates exported XML files in the install directory, and can be found using the following path:

```
$(INSTALL_HOME)/db/export/tables/<tablename>.xml
```

Where *<tablename>* specifies the name of the extracted database table.

dbImport

The dbImport utility imports policies into the database.

Purpose

The dbImport utility imports policies into the database. The utility creates the XML data sets as a result of previously running the dbExport utility.

The data that the dbImport utility imports must be placed in the following install directory:

```
$(INSTALL_HOME)/db/export/tables
```

This is the same location where the dbExport utility creates the exported XML files.

Syntax

To invoke the dbImport utility, use the following syntax:

```
./dbImport
```

Tables exported

By default, the dbImport utility imports the following tables into the database:

- POLICY
- POLICYTOALARMACTION
- POLICYTOEMAILACTION
- POLICYTORULE
- RULE
- RULEBLOB
- RULETODEFINITION
- RULETOCOMMANDSETACTION
- DEFINITION
- EVALUATION
- COMMANDSETACTION
- EMAILACTION
- ALARMACTION
- GLOBALPARAMETER
- EXTRACTION
- PARAMETER-GROUP
- PARAMETERGROUPVALUE

Output

The `dbImport` utility imports the data sets to the database tables as specified.

Security on imports

The imported tables are automatically created in the realm listing under the **Realm Access Control Tab** in Compliance Security Administration. By default, there is no realm security. Users do not have the ability to view realms. To enable view access and execution of the imported policies within the realms, security permissions must be applied. For further information on applying security in Realm Management, see the *ITNCM User Guide*.

policyExport

The `policyExport` utility extracts policies from the database.

Purpose

The policy extraction process entails the export of all data associated with a policy, including rules, definitions, extractions, actions and parameters.

Syntax

When you invoke the `policyExport` utility, the following syntax is supported:

```
---------------------------------------------------------------
1: ./policyExport.sh -policy <name>=<revision>
2: ./policyExport.sh -policy <name>=<revision>,<name>=<revision>
3: ./policyExport.sh -policy <path>
4: ./policyExport.sh -policy <path>,<path>
5: ./policyExport.sh -all
---------------------------------------------------------------
Optional: -filename <exported zip name>
---------------------------------------------------------------
```

Note:

If `-filename` is not supplied to any of the commands, then a zip file of the following format will be created:

For a multiple policy export:
`policies_<date/time>.zip`

For a single policy export
`policy_<date/time>.zip`

Note: All zip files exported will be created in:
`<install_dir>/compliance/db/export/policies`
where *`install_dir`* is your installation directory.

Examples

The following examples demonstrate how this syntax is specified.

`./policyExport -policy newPolicy=1`
Where *`newPolicy`* is the name of the policy and *`1`* is the revision.

`./policyExport -realm NSA/v1 -filename nsa_policies`
Where *`NSA/v1`* is the name of a policy realm that you want to export and *`nsa_policies`* is the name of the exported zip file.

All policies in this realm and subrealms will be exported and saved as:
`<install_dir>/compliance/db/export/policies/nsa_policies.zip`

Where *`install_dir`* is your installation directory.

`./policyExport -all`
Where *`-all`* means that all policies in the database will be exported.

The policies will be saved as:
`<install_dir>/compliance/db/export/policies/policies_<date/time>.zip`

Where *`install_dir`* is your installation directory, and *`date/time`* is determined by the date and time the policies are exported.

policyImport

The `policyImport` utility imports the data of previously exported policies back into the database.

Purpose

The `policyImport` utility reads in the filename of a previously exported zip or single policy xml file and writes the policy information into the database.

Syntax

When you invoke the `policyImport` utility the following syntax is supported:

```
----------------------------------------------------------------
1: ./policyImport.sh <filename>
----------------------------------------------------------------
Optional: -overwrite, suppressWarnings
----------------------------------------------------------------
```

Note: The following commands are optional:

-overwrite
This option forces an overwrite of an existing policy component with the same version.

-suppressWarnings
This option suppresses all warning messages displayed on the screen.

Note: Files to be imported must exist in the following directory and be a zip file or single policy xml file:
`$(INSTALL_HOME)/compliance/db/export/policies`

Example

The following command will attempt to import all policies in the zip file:

```
./policyImport.sh NSA_policies.zip
```

Note: If a policy or policy parameter already exists in the system it will not be imported unless the `-overwrite` option in specified.

The `policyImport` utility imports the data sets to the database tables as specified.

Security on imports

The imported tables are automatically created in the realm listing under the **Realm Access Control Tab** in Compliance Security Administration. By default, there is no realm security. Users do not have the ability to view realms. To enable view access and execution of the imported policies within the realms, security permissions must be applied. For further information on applying security in Realm Management, see the *IBM Tivoli Netcool Configuration Manager User Guide*.

Administering BulkLoader

To import network resources into the database, use the `BulkLoader` utility. You execute the `BulkLoader` utility by specifying the `BulkLoader` keyword when executing the `icosutil` utility.

BulkLoader utility overview

The `BulkLoader` utility populates the database with a set of resources, so that normal configuration management tasks can begin in ITNCM - Base.

The `BulkLoader` utility reads a CSV file that contains a list of network resources to import. It then uses the API to create placeholders for each resource and imports the current configuration of each. See the *ITNCM API Guide* for information about how the API creates resources and imports configurations.

Note: When using the `BulkLoader` and SSH, the Resource Access Document (RAD) needs to be predefined at the realm level.

Note: Use the `BulkLoader` utility only when ITNCM - Base is stand alone. Do not use the `BulkLoader` utility when integrated with ITNM.

Update Sequence

When invoked, the `BulkLoader` utility selects a record from the specified CSV file and reads the database, matches on the host name and VTMOS, and if found, updates the access method(s). If the host name is not found in the database, the `BulkLoader` utility attempts to match on realm. If the sub-realm under existing root realm is not found, and if the `-cr` parameter is specified, then only it is added. Next the `BulkLoader` utility adds the resource to the database and then attempts to import the resource.

The significance of this sequence (update if present and add if not present) means you can use the `BulkLoader` utility to make security access changes to multiple devices that are already present in the database.

Data file formatting rules

The accepted format for the file is CSV, in which each row provides resource information. The data file imported by the `BulkLoader` utility must comply with the following format:

```
NCMrealm,10.219.1.34,Cisco,Router,26*,12.2-*,"telnet,none,true,testuser1,
password1,enable1,none,23","ssh,sshtype1,true,testuser1,password1,
enable1,2600e,22"NCMrealm,10.219.1.35,Cisco,Router,26*,12.2-*,
"telnet,none,true,testuser2,password2,enable2,none,23","ssh,sshtype2,
true,testuser2,password2,enable2,2600d,22"NCMrealm,10.219.1.36,Cisco,
```

```
Router,26*,12.2-*,"telnet,none,true,testuser3,password3,enable3,none
,23","ssh,sshtype3,true,testuser3,password3,enable3,2600c,22"
```

To import multiple network devices the 'range' keyword must be used. In the following example all host names between 10.219.34.1 and 10.219.34.21 are specified:

```
NCMrealm,range:10.219.34.1-10.219.34.21,Cisco,Router,26*,12.2-*,
"telnet,none,true,testuser,password,enable,none,23",
"ssh,sshtype,true,testuser,password,enable,2600e,22"
```

Or alternatively, the range can be specified using a subnet suffix:

```
NCMrealm,range:10.219.34.1/24,Cisco,Router,26*,12.2-*,"telnet,
none,true,testuser,password,enable,none,23","ssh,sshtype,true,
testuser,password,enable,2600e,22"
```

It is critical that the VTMOS be properly defined for the BulkLoader entry to match the support list for that device. Otherwise, you will invoke autodiscovery and no device-level RAD will be created. The following list provides the rules for specifying VTMOS:

- If a specific device is given you must have values for VTMOS.
- If a range of IP addresses is used you can use an `'*'` to specify VTMOS.

The following list provides the rules for access data:

- The date file for the import must follow the format specified.
- Optional: If the access type is not specified, the default access type is used.
- No access type specification is required for importing with a range of IP addresses.

Note: The field in the RAD `socketConnectTimeout` provides the connect timeout in milliseconds. This is used to control the socket connection timeout for Auto-Discovery.

CSV file description

ITNCM - Base supports fallback methods to communicate with each resource, as well as different user names and passwords on a per resource basis. All remaining data in this table are optional.

If login credentials are not included for a particular device, the `BulkLoader` utility will use authentication or RAD objects already present. The utility will also default to using telnet as the communication protocol with no fallback access method.

ITNCM - Base provides three access methods: TELNET, SSH, and alt-telnet or console. The order of the attempts is defined on a left to right basis within the data file.

The following table describes the information that must be contained within the CSV file:

Column	Description
realm	Specifies the location where the resource will reside in the database. The realm is defined by creating a path name that would exist under the main realm that was created during the ITNCM - Base installation process. This is a required field.
host name	Specifies the name of the resource being imported. This name must be resolved by a DNS server or it must be logged in the host file for the server. ITNCM - Base will use the host name for device communication. This is a required field.
vendor	Specifies the vendor name for the resource. This is a required field. **Note:** For the VTMOS fields, the system validates that you have provided supported values, but you still must be sure that the values are accurate for the resource you are importing.

Column	Description
type	Specifies the type of resource (router, switch, or firewall). This is a required field.
model	Specifies the model number of the resource. This is a required field.
OS	Specifies the version of OS running on the resource. This is a required field.
access-type	Specifies a title for the access method; it is just a descriptor placed in the access script that is defined for the resource. If the access-type defined in the data file is not the same as the default access script, the resource will use the new method.
ssh-type	Specifies a flag that determines the method used to access the resource. Select one of the following options: • `none` — Uses TELNET. • `ssh1` — Uses the des cipher, which is acceptable for CISCO resources. • `ssh2` — Uses blowfish cipher, which is acceptable for Juniper resources.
streaming	Specifies a flag that indicates whether streaming data will be used. Select one of the following options: • `True` — Streaming data will be used. • `False` — ftp and/or tftp will be used.
username	Specifies the username that will be used to log onto the resource. This username needs to belong to a group with appropriate privileges to modify the resource.
password	Specifies the password associated with the username from the previous column.
enable password	Used for resources to allow modifications to the resource.
alt-hostname	Specifies an optional parameter that is used for access to the device through a console server.
port	Specifies the port that will be used to communicate with the resource.

Each column must be separated by a comma. Values can be enclosed in single or double quotes. All values that are not quoted must be composed of letters (lower or uppercase), numbers, dollar signs ($), periods (.), parentheses, dashes (-), and underlines (_). Comment lines begin with a # (pound sign).

The text file may use any other characters, but the entire value that includes the non-supported character must be enclosed in single or double quotes.

The last eight security access information values (starting with com-type) may be repeated, which is indicated by the double quote marks in the header row. Being repeatable means that access information for both SSH and telnet can be entered within the same record.

If a value does not exist for one of the fields, make sure to use a space between the commas. When viewed in a spreadsheet, each set of eight repeating values will occupy a single cell, as shown in the following example:

```
com-type,ssh-type,streaming,username,password,enable-password,alt-hostname,port
telnet,none,true,go,go,go, ,23
telnet,none,true,go,go,go, ,23
telnet,none,true,go,go,go, ,23
telnet,none,true,go,go,go, ,23
```

```
telnet,none,true,go,go,go, ,23
telnet,none,true,go,go,go, ,23
```

Client setup

To run the `BulkLoader` utility from a client machine, the client setup procedures as described in the *ITNCM Installation Guide* must be performed for the operating system that is running on the client machine.

Syntax

```
icosutil BulkLoader {required parameters) (optional parameters}
```

Parameters

`BulkLoader`
Specifies the keyword that instructs the `icosutil` utility to run the `BulkLoader` utility.

`-u`
Specifies a required option when using the `Encrypt` keyword.

required parameters
Specifies one of the following required parameters:

Required parameter	Description
`-l`	Specifies a valid username.
`-p`	Specifies the password associated with the username being used. This should be clear text.
`-f`	Specifies the path and file name of the data file containing the list of resources to import.

optional parameters
Specifies one of the following optional parameters:

Optional parameter	Description
`-server`	Specifies the hostname or IP address of the ITNCM - Base server to which you are connecting.
`-cu[realm]`	Specifies a command that enables network resources to be created where the VTMOS = unknown/unknown/unknown/unknown. This is in the event that the import fails. By default it will create the network resource in the realm specified in the CSV for the import. If a realm is specified, the unknown Network Resource will instead be created in the realm.
`-port`	Specifies the port number to use. 16310 is the default port for the ITNCM - Base server, but this can be changed at installation time. This user must belong to a group with add rights to the realm in which you are adding the resources, as well as add rights to resources in that realm.
`-cr`	Specifies that you want to create the realm (under existing root realm), if the realm you are importing to does not already exist under the existing root realm.

Optional parameter	Description
-poll	Specifies import status polling
–help	Specifies usage information about the BulkLoader utility.
-update	Updates the existing RAD.

Sample

The following example specifies a sample command line for the BulkLoader utility:

```
icosutil BulkLoader -server 1.2.3.4 -port 80 -l name -p pwd -f /home/icosuser/
list.csv -cr
```

The icosutil utility would return the following:

Auto discovery

The BulkLoader utility provides an autodiscovery function that loads devices even when you are not entirely sure what is on your network. The utility “discovers” the Vendor, Type, Model, and Operating System (VTMOS) of each host and loads it into ITNCM - Base.

The following example shows how to use the autodiscovery function that the BulkLoader utility provides:

	A	B	C	D	E	F	G	H
1	#Realm	Hostname	Vendor	Type	Model	OS	Access Method 1 (optional)	Access Method 2 (optional)
2	IPX01-customer-A	sales_lab_2600-2	Cisco	Router	26*	*12.2*	telnet,none,true,goo, go,go,none,23	telnet,none,true,cisco, cisco,cisco,none,23
3	IPX01-customer-A	sales_lab-7	Cisco	Router	26*	*12.2*		
4	IPX04-customer-B	range:10.217. 200.1/24	*	*	*	*		
5	IPX04-customer-B	range:10.219. 34.10-10.219. 34.20	*	*	*	*		

The following list describes the example:

- Sales_lab_2600-2 will be imported as a Cisco/Router/26* regardless of its true type. It uses custom primary and secondary access methods.
- Sales_lab-7 will be discovered using standard authentication resource with values for VTMOS. Primary and secondary access methods are not required.
- The 10.217.200.1 subnet will be discovered using standard authentication resource and VTMOS as ‘*'. Primary and secondary access methods are not required.
- Hosts 10.219.34.10 through 10.219.34.215 will be discovered using standard authentication methods. The system will find the access method for each device it attempts to import.

Results of the BulkLoader utility

After you invoke the utility, the program parses the text input file you are passing in. If any syntax errors are found, the program will return the input line in which the error was found. It then continues processing the records from the data file.

The following example shows the results of a syntax error. In this case, the Vendor name is incorrect.

```
Invalid Vendor, Type, Model or OS: 2600f/Cico/Router/26*/12.2-* - On line 3
```

If a sub-realm name in text file cannot be found, that sub-realm will be created under the given root realm, as long as you specified the `-cr` attribute when invoking the command. If the sub-realm does not exist under the given root realm and you did not use the `-cr` option, then the `BulkLoader` utility returns the following error:

```
Realm ITNCM/realm1 not found skipping: 2600e/Cisco/Router/26*/12.2-* - On line 2
```

If the resource being imported already exists in the ITNCM - Base database, the resource is updated with the VTMOS information from the data file.

```
Updating: 2600b/Cisco/Router/26*/12.2-* - On line 4
```

If the input file is in the correct format, a unit of work (UOW) consisting of an import is submitted for each line of the text file, meaning each resource is imported as a separate UOW. Each UOW will automatically begin processing unless approvals are required first. If approvals are required, you won't be returned to the command prompt until after each UOW has been approved.

When finished, the utility shows a summary that includes the number of sucessful and failed import submittals.

```
Importing: 2600c/Cisco/Router/26*/12.2-* - On line 2
Updating: 2600k/Cisco/Router/26*/12.2-* - On line 3
Updating: 2600h/Cisco/Router/26*/12.2-* - On line 4
Imported: 2600c/Cisco/Router/26*/12.2-* : Result - SUCCESS
```

Housekeeping

Use this information about Netcool Configuration Manager to understand how to configure housekeeping and how record removal affects the database.

About Compliance housekeeping

Housekeeping refers to the tasks associated with freeing up storage space and improving run-time performance.

As increasing numbers of network resources and configurations per resource are added onto the Netcool Configuration Manager - Compliance application, housekeeping duties become particularly important to free up storage space and to maximize run time performance. This section promotes an understanding of the housekeeping functionality.

Record removal

Housekeeping tasks such as removing records from the database causes the removal of the following:

- Record results from the results database tables
- All result associations
- Policy validation
- Remedial queue

Thus, once a record is removed from the database, there is no trace of the records left in the database tables, and they cannot be retrieved. Each record removal request takes on average 90 seconds to complete. However, depending on the number of records to be deleted, this removal request may take longer. Before removing records from the database, consider the following:

Aging periods
Record deletion is configurable using the parameters that the housekeeping utility provides. One parameter is how long records have been on the system. Valid values are weekly, monthly, quarterly,

twice a year, and yearly. For example, if monthly is chosen, the housekeeping utility removes all records associated with the specified process results that are a month old or older.

Number of results

Another record deletion parameter is the number of records between two and ten to keep in the process results. The minimum number of records is two in order to ensure that there is always at least two results left for comparison reasons. This means that all records associated with the specified process results will be deleted, except the most recent x records.

However, some process results have immunity to this rule. For example, a user may specify that at least four records associated with the specified process results be retained for each process. After the house keeping utility has been run, there may be seven results remaining. The reason being that three of the results may have a state of `Pending Approval`, `Ready To Execute`, and `Pending on ITNCM - Base`. Process results with any of these states are excluded from the housekeeping operation.

Database tables used in housekeeping

The key database tables used by the housekeeping utility are summarized in the following table:

Table 7. List of key housekeeping database tables

Database table name	Description
SystemOptions	Details of when processResults should be deleted. In the case of normal processes, also details the minimum number of results that should be left in the database.
ProcessResult	Details the start time of a process result, the process state (Finished, scheduled, and so forth), and the execution type (Unscheduled, OOB, AdHoc, and so forth).
CurrentDevice PolicyResult	Details of all current device policies, used to determine if the results can be removed. Please refer to removal criteria.
RemedialQueue	Once the current results are checked, the remedial queue will be examined.

Removal criteria

There are a number of removal criteria that determine if the records associated with the specified process results can be deleted. The following list identifies instances where a record associated with a specified process result will NOT be deleted from the database tables:

- If the process result finishes AFTER the deletion time indicated. For example, the process result has a state of `Scheduled` when the housekeeping takes place.
- If the process result does not have a `Finished` or `Error` state. For example, the process result has a status of `Queued` for execution.
- If the process result `ck` is in the CurrentDevicePolicyResult table.
- If the process result is in the remedialQueue table with the status of `Pending Approval`, `Ready to Execute`, or `Pending on ITNCM - Base`.
- If the number of process results left in the database after the deletion would leave the results below the number required by the user (not applicable for adhoc or OOB processes). In this case, the most recent process results that fulfil the deletion criteria will be saved until the amount left is adequate.

If the process result fulfils the deletion criteria, all its results will be deleted from the following tables:

- processResult
- processResultToPolicyResult
- policyResult
- policyResultToRuleResult
- ruleResult

- ruleResultToDefResult
- ruleResultToCorractResult
- defResult
- correctiveActionResult
- defResultToEvalResult
- EvaluationResult
- policyValidationSummary
- remedialQueue

Methods to run the housekeeping utility

The housekeeping administrator can configure housekeeping in the **WorkFlowManager.properties** file, or alternatively run the housekeeping utility through the User Interface or by setting up a cron job.

Setting the DB2 page size

The following configuration settings are recommended for Netcool Configuration Manager housekeeping:

```
db2 update db cfg using logfilsiz 5000
db2 update db cfg for itncm using logprimary 200
db2 update db cfg for itncm using logsecondary 50
```

Note: Use the following information to set the transaction log file size:

- To clear 50000 UOWs during housekeeping, set the transaction log file size to approximately 5000.
- To clear 100000 UOWs during housekeeping, set the transaction log file size to approximately 16384

Tip: You can use the db2autoconfigure utility to auto-configure a number of DB2 configuration settings, such as the DB2 transaction log file size.

Tip: You can use the DB2 Activity Monitor wizard to help you determine the levels to increase it to. For more information, see the DB2 documentation.

The following configuration step is recommended to reduce errors in compliance execution:

```
update db cfg for your_itncm using LOCKLIST
8192
```

Set the DB2 page size to 32768.

For example (database creation example):

```
db2 create database itncm automatic storage yes pagesize 32768 dft_extent_sz 32
```

Configuring the removal of records

Users can configure the housekeeping options using the WorkFlowManager.properties file. For more information, see *Configuring the removal of records* in the *IBM Tivoli Netcool Configuration Manager Administration Guide*.

Removing records using the GUI

Remove records from the results database tables by using the House Keeping utility. Access the House Keeping utility by using the House Keeping Options screen. For more information, see *Removing records using the GUI* in the *IBM Tivoli Netcool Configuration Manager Administration Guide*.

Viewing the compliance event log

The ITNCM-Compliance event log contains general information about the operations that occur within the application. View the event log by accessing the User Audit Trail from the User Interface. For more information, see *Viewing the compliance event log* in the *IBM Tivoli Netcool Configuration Manager Administration Guide*.

Configuring the removal of records

Users can configure the housekeeping options using the `WorkFlowManager.properties` file.

You must be familiar with the command line and a text editor such as vi to perform this task.

To configure housekeeping using the `WorkFlowManager.properties` file, follow these steps.

1. Log onto the ITNCM-Compliance server as the specified ICOSUSER.
2. Edit the following file: `/opt/IBM/tivoli/netcool/ncm/compliance/config/properties/WorkFlowManager.properties`.
3. Set `houseKeepingEnabled` to `true` to enable housekeeping.
4. Set the time at which housekeeping is to be run by using the `houseKeepingStartHour` property.

   The `houseKeepingStartHour` property is in Greenwich Mean Time (GMT).

   The default is 5.
5. Save the changes made to the properties file, and exit.

`houseKeepingStartHour=9` will execute housekeeping at 9am GMT each day.

You can also remove records from the results database tables by using the House Keeping Options screen.

Removing records using the GUI

Remove records from the results database tables by using the House Keeping utility. Access the House Keeping utility by using the House Keeping Options screen.

You can only perform this task if you are the ITNCM-Compliance Application Administrator. See the *ITNCM User Guide* for more information.

To remove records from the results database tables, follow these steps.

1. From the User Interface, select Admin | House Keeping options. The House Keeping Options screen is displayed. The following table describes each of the fields in the screen.

Option	**Description**
Screen item	**Description**
Process Results:	Specifies that all records associated with the specified process results be deleted from the results database tables. The process itself will not be deleted.
Adhoc Process Results:	Specifies that all records associated with the specified adhoc process results be deleted from the results database tables. The adhoc process itself will not be deleted.
Automated Process Results:	Specifies that all records associated with the specified auto-initiated process results be deleted from the results database tables. The auto-initiated process itself will not be deleted.

2. From each of the drop down lists (**Process Results:**, **Adhoc Process Results:**, **Automated Process Results:**, and **Adhoc Process Results:**) select the frequency of record deletion: Weekly, Monthly, Quarterly, Twice a year, or Yearly.
3. From the **Keep at least:** drop down list, select the number of records to keep (from two to ten) in the process results.
4. Click Apply to accept the specified options. Or, click Cancel to cancel the specified options.

You can also remove records from the results database tables by configuring housekeeping in the **`WorkFlowManager.properties`**.

Viewing the compliance event log

The ITNCM-Compliance event log contains general information about the operations that occur within the application. View the event log by accessing the User Audit Trail from the User Interface.

To view the event log, follow these steps.

From the User Interface, select Admin | User Audit Trail. The event log is displayed. The following table describes each of the fields in the event log.

Option	Description
Screen item	Description
Time	Specifies the time in which the event took place. All times are server side.
Description	Provides a description of the event.
User	Specifies the username of the person who triggered the event.

The following shows examples of events in the event log:

```
Time                                    Description                                    User

07-Jul-2011                          User admin has logged in            admin
07-Jul-2011                          User admin has logged out        admin
```

Archiving IDT session logs (IDTArchive)

To archive IDT session logs from ITNCM - Base network resources, specify the `IDTArchive` keyword when executing the `icosutil` utility. This keyword executes the `IDTArchive` utility with the options specified on the command line. One of these options is a properties file that allows you to specify the criteria to use when archiving IDT session logs.

Syntax

```
./icosutil IDTArchive -f /opt/IBM/tivoli/netcool/ncm/config/properties/
IDTSessionArchivingUtility.properties
```

Description

The `IDTArchive` utility archives IDT session logs from ITNCM - Base network resources based on criteria specified in the `IDTSessionArchivingUtility.properties` file. By default, the `IDTSessionArchivingUtility.properties` file resides in the `/opt/IBM/tivoli/netcool/ncm/config/properties` directory.

The `IDTArchive` utility can retain or delete IDT session logs based on the following criteria:

- `firstNEntries`
- `lastNEntries`
- `startDate`
- `endDate`
- `startDateFromCurrentDateUnits`
- `startDateFromCurrentDateUnitOfTime`
- `endDateFromCurrentDateUnits`
- `endDateFromCurrentDateUnitOfTime`

Examples of usage are available in the `IDTArchive` utility.

Upon completion, the IDTArchive utility creates an XML file and a text file in the default directory /opt/IBM/tivoli/netcool/ncm/bin. The IDTArchive utility uses the following naming scheme:

```
IDT_TIMESTAMP.xml

IDT_<TIMESTAMP>.txt
```

Where: *TIMESTAMP* specifies the date and time in which the IDTArchive utility created the files.

Parameters

IDTArchive
Specifies the keyword that instructs the icosutil utility to execute the IDTArchive utility. The IDTArchive utility proceeds to archive IDT session logs from ITNCM - Base network resources based on criteria specified in the IDTSessionArchivingUtility.properties file.

-f
Specifies an option when using the IDTArchive keyword.

IDTSessionArchivingUtility.properties
Specifies the IDTSessionArchivingUtility.properties file that defines the criteria that the IDTArchive utility uses to archive IDT session logs from ITNCM - Base network resources. The IDTSessionArchivingUtility.properties file allows you to define the following criteria:

- Server hostname
- Port number
- Login credentials for the server on which IDT is installed
- Configurable parameters to use when archiving IDT session logs

Multiple selection criteria may be used for determining which session logs to keep or delete.

Note: To archive IDT session logs using the previously listed utilities, the user must belong to a group with the IDT Administration activity.

Archiving and deleting versioned configurations from the database (ConfigArchive)

To archive and/or delete versioned configurations from the database, specify the ConfigArchive keyword when executing the icosutil utility. This keyword executes the ConfigArchive utility with the options specified on the command line. One of these options is a properties file that allows you to specify which versioned configurations to delete from the database.

Syntax

```
./icosutil ConfigArchive -f /opt/IBM/tivoli/netcool/ncm/config/properties/
configArchivingUtility.properties
```

Description

The ConfigArchive utility archives and/or deletes versioned configurations from the database based on criteria specified in the configArchivingUtility.properties file. By default, the configArchivingUtility.properties file resides in the /opt/IBM/tivoli/netcool/ncm/config/properties directory.

Fix Pack 1 The ConfigArchive utility is used to manage filtered versions of configurations, and does not work with configurations which are current, draft or marked as 'golden'.

The ConfigArchive utility can retain or delete versioned configurations based on the following criteria:

- SelectedConfigurationAction

- `lastNEntries`
- `endDateFromCurrentDateUnits`
- `endDateFromCurrentDateUnitsOfTime`

If archiving is enabled, one gzipped xml archive file is created for each versioned configuration that is archived. The ConfigArchive utility uses the following naming scheme:
`CONFIG_<internal_config_version_id>_<`*TIMESTAMP*`>.xml.gz`

Where: *TIMESTAMP* specifies the date and time in which the ConfigArchive utility created the Archive file.

Parameters

`ConfigArchive`
Specifies the keyword that instructs the `icosutil` utility to execute the ConfigArchive utility. The ConfigArchive utility proceeds to delete versioned configurations from the database based on criteria specified in the `configArchivingUtility.properties` file.

`-f`
Specifies an option when using the `ConfigArchive` keyword.

`configArchivingUtility.properties`
Specifies the `configArchivingUtility.properties` file that defines the criteria that the ConfigArchive utility uses to delete versioned configurations from the database.

Notes

Note: Fix Pack 1 Golden configurations cannot not be archived.

As increasing numbers of network resources and configurations per resource are added to Netcool Configuration Manager - Base, housekeeping duties become particularly important to free up storage space and maximize run time performance. The following utilities perform these housekeeping duties:

ConfigArchive
Archives and deletes versioned configurations from the database.

WorkHousekeeping
Deletes completed UOWs from the database.

For more information, see “Deleting completed UOWs from the database (WorkHousekeeping)” on page 123.

Archive
Archives the UOW.

For more information, see “Archiving a UOW (Archive)” on page 125.

Note: In order to perform Housekeeping duties using the previously listed utilities, the user must belong to a group with the Housekeeping activity.

To delete a versioned configuration, the user must also have delete rights for the contents of the realm containing the configuration.

Using the ConfigArchive utility

The ConfigArchive utility can be run concurrently with other units of work. Other UOW types will not encounter work conflict warnings, or device already locked errors.

Follow these steps to delete versioned configurations from the database:

1. Edit the sample `configArchivingUtility.properties` file distributed with the Netcool Configuration Manager - Base software. This sample properties file demonstrates how the ConfigArchive utility works, and comes complete with examples. The `configurationHousekeepingUtility.properties` file can be configured to run with your system.
2. Run the ConfigArchive utility using the following command:

```
./icosutil ConfigArchive -f /opt/IBM/tivoli/netcool/ncm/config/properties/
configArchivingUtility.properties [-Xms] [-Xmx]
```

Where:

Xms
Sets the minimum size of the memory allocation pool (minimum heap size). Set this value to a multiple of 1024 that is greater than 1 MB. As a general rule, set minimum heap size (`-Xms`) equal to the maximum heap size (`-Xmx`).

Xmx
Sets the maximum Java heap size. Set this value to a multiple of 1024 that is greater than 1 MB.

3. Depending on whether the utility is being run with the Report-Only flag set to true or false, the utility will generate a total of the configurations to be deleted, or it actually removes the selected configurations. Either way, the report is generated.
4. 5. The archiveConfigFlag in `configArchivingUtility.properties` controls whether versioned configurations will be archived prior to deletion. The default value (true) delivered with the `configArchivingUtility.properties` file will mean Archive files will be created.
5. Review the report produced to verify that all expected configurations were archived or deleted.

Results

A `ReportOnly` option is available in the `configArchivingUtility.properties` file. If running with the `ReportOnly` option set to False, all versioned configurations that meet the selection criteria will be deleted completely from the database, Queue Manager, Resources tab, and the Reports tab.

If a configuration selected for deletion cannot be deleted due to insufficient security rights, the utility will cease to run. The utility can only work if the user has housekeeping privileges.

If running with the `ReportOnly` option set to True, a report will be generated that shows the number of configurations that will be deleted.

This flag specifies whether or not the utility is being executed in `ReportOnly` mode. If the value is true, the configuration housekeeping utility report will show how many configurations would have been deleted based upon the criteria, but the configs will not actually be deleted (this is useful for testing new criteria settings). If the value is false, the report is generated and the configs are permanently deleted from Netcool Configuration Manager - Base. The `ReportOnly` default value is true.

By default the ‘archive root’ directory that archive files will be written to is the `$INSTALL_DIR/configurationArchive` directory, but this can be overridden by specifying the full path to another directory (to use as the ‘archive root’ directory) in the ‘archiveDirectory’ property of the `configArchivingUtility.properties` file. Under the ‘archive root’ directory archive files will be written to a subdirectory corresponding to a devices realm or subrealm.

In addition to the versioned configuration information the archive file will also contain some resource level information (‘RESOURCEBROWSER’ section in the xml archive file), this is to allow users to more easily associate the archived configuration information with the relevant resource.

Restoring a Versioned Configuration from an archive (ConfigRestore)

To restore a versioned configuration from an archive, specify the ConfigRestore keyword when executing the icosutil utility. This keyword executes the ConfigRestore utility with the options specified on the command line. One of these options is the properties file (`configArchivingUtility.properties`).

Note: To avoid possible corruption issues, only the original unchanged archive files can be loaded into the system using the ConfigRestore utility.

Syntax

```
./icosutil ConfigRestore -f /opt/IBM/tivoli/netcool/ncm/config/properties/
configArchivingUtility.properties -archiveName gzipped_xmlfilename
```

Description

The ConfigRestore utility restores the specified versioned configuration from an archive. The ConfigRestore utility pulls the required information from the gzip XML file specified in gzipped_xmlfilename and restores the versioned configuration using the ITNCM Server criteria specified in the workArchivingUtility.properties file. By default, the workArchivingUtility.properties file resides in the /opt/IBM/tivoli/netcool/ncm/config/properties directory.

Restored configuration versions are presented in the GUI in the exact same state as the configuration version had when it was archived. Configuration versions are restored with same internal identifier values as the archived configuration had, this means it is not suitable for exporting/importing to different installations.

In order to perform Archive restoring, the user must belong to a group with the Manage Archive activity.

Parameters

ConfigRestore
Specifies the keyword that instructs the icosutil utility to execute the ConfigRestore utility. The ConfigRestore utility proceeds to restore the versioned configuration.

-f
Specifies an option when using the ConfigRestore keyword.

configArchivingUtility.properties
Argument for the 'f' option. Specifies the configArchivingUtility.properties file that defines the ITNCM Server criteria that the ConfigRestore utility uses to restore versioned configuration from an archive.

-archiveName
Specifies the name of the archive that contains the versioned configuration that you want to restore. The location of the archive does not have to be specified. Information about the location of the archive is stored(at time of archiving) internally in ITNCM. If the archive is not in the location stored internally in ITNCM then the ConfigRestore utility will attempt to load the archive from the current executing directory.

gzipped_xmlfilename
Specifies the name of the gzip XML file that the ConfigRestore utility uses to pull the required information.

Deleting completed UOWs from the database (WorkHousekeeping)

To delete completed UOWs from the database, specify the `WorkHousekeeping` keyword when executing the `icosutil` utility. This keyword executes the `WorkHousekeeping` utility with the options specified on the command line. One of these options is a properties file that allows you to specify which completed UOWs to delete from the database.

Syntax

```
./icosutil WorkHousekeeping -f /opt/IBM/tivoli/netcool/ncm/config/properties/
workHousekeepingUtility.properties
```

Description

The `WorkHousekeeping` utility deletes completed UOWs from the database based on criteria specified in the `workHousekeepingUtility.properties` file. By default, the `workHousekeepingUtility.properties` file resides in the `/opt/IBM/tivoli/netcool/ncm/config/properties` directory.

For each UOW that is deleted, the `WorkHousekeeping` utility also deletes the corresponding native command files stored on the server. When this option is executed, a text file is produced in the running directory.

The `WorkHousekeeping` utility can retain or delete completed UOWs based on the following criteria:

- firstNEntries
- lastNEntries
- startdate
- enddate

Upon completion, the WorkHousekeeping utility creates a text file in the running directory. The WorkHousekeeping utility uses the following naming scheme:

```
WorkHouseKeeping_<TIMESTAMP>.txt
```

Where: *TIMESTAMP* specifies the date and time in which the WorkHousekeeping utility created the files.

Parameters

WorkHousekeeping
Specifies the keyword that instructs the icosutil utility to execute the WorkHousekeeping utility. The WorkHousekeeping utility proceeds to delete completed UOWs from the database based on criteria specified in the workHousekeepingUtility.properties file.

-f
Specifies an option when using the WorkHousekeeping keyword.

workHousekeepingUtility.properties
Specifies the workHousekeepingUtility.properties file that defines the criteria that the WorkHousekeeping utility uses to delete completed UOWs from the database.

Notes

As increasing numbers of network resources and configurations per resource are added to ITNCM - Base, housekeeping duties become particularly important to free up storage space and maximize run time performance. The following utilities perform these housekeeping duties:

Fix Pack 3

ConfigArchive
Deletes versioned configurations from the database.

For more information, see "Archiving and deleting versioned configurations from the database (ConfigArchive)" on page 120

WorkHousekeeping
Deletes completed UOWs from the database.

Archive
Archives the UOW.

For more information, see "Archiving a UOW (Archive)" on page 125.

Note: In order to perform Housekeeping duties using the previously listed utilities, the user must belong to a group with the Housekeeping and View All Work activity.

Using the WorkHousekeeping utility

The WorkHousekeeping utility can be run concurrently with other units of work. Other UOW types will not encounter work conflict warnings, or device already locked errors. This is because the config housekeeping process is running simultaneously.

Follow these steps to delete versioned configurations from the database:

1. Edit the sample workHousekeepingUtility.properties file distributed with the ITNCM - Base software. This sample properties file demonstrates how the WorkHousekeeping utility works, and comes complete with examples. The workHousekeepingUtility.properties file can be configured to run with your system.
2. Run the WorkHousekeeping utility using the following command:

```
./icosutil WorkHousekeeping -f /opt/IBM/tivoli/netcool/ncm/config/properties/
workHousekeepingUtility.properties [-Xms] [-Xmx]
```

Where:

- `-Xms` — Sets the minimum size of the memory allocation pool (minimum heap size). Set this value to a multiple of 1024 that is greater than 1 MB. As a general rule, set minimum heap size (`-Xms`) equal to the maximum heap size (`-Xmx`).
- `-Xmx` — Sets the maximum Java heap size. Set this value to a multiple of 1024 that is greater than 1 MB.

3. Depending on whether the utility is being run with the Report-Only flag set to true or false, the utility will generate a list of the UOWs to be deleted, or it actually removes the selected UOWs. Either way, the report is generated.
4. Review the report produced to verify that all expected UOWs were deleted.

Results

A `ReportOnly` option is available in the `workHousekeepingUtility.properties` file. If running with the `ReportOnly` option set to False, all completed UOWs that meet the selection criteria will be deleted completely from the database, Queue Manager, Resources tab, and the Reports tab.

If a UOW selected for deletion cannot be deleted due to insufficient security rights, the utility will cease to run. The utility can only work if the user has housekeeping privileges.

If running with the `ReportOnly` option set to True, a report will be generated that shows the number of completed UOWs that will be deleted.

This flag specifies whether or not the utility is being executed in `ReportOnly` mode. If the value is true, the work housekeeping utility report will show how many UOWs would have been deleted based upon the criteria, but the UOWs will not actually be deleted (this is useful for testing new criteria settings). If the value is false, the report is generated and the UOWs are permanently deleted from ITNCM - Base. The `ReportOnly` default value is true.

Archiving a UOW (Archive)

To archive a UOW, specify the `Archive` keyword when executing the `icosutil` utility. This keyword executes the `Archive` utility with the options specified on the command line. One of these options is a properties file that allows you to specify which UOWs to archive.

Syntax

```
./icosutil Archive -f /opt/IBM/tivoli/netcool/ncm/config/properties/
workArchivingUtility.properties
```

Description

The `Archive` utility archives UOWs from ITNCM - Base, based on criteria specified in the `workArchivingUtility.properties` file. By default, the `workArchivingUtility.properties` file resides in the `/opt/IBM/tivoli/netcool/ncm/config/properties` directory.

The `Archive` utility can archive UOWs based on the following criteria:

- `firstNEntries`
- `lastNEntries`
- `startdate`
- `enddate`

Note: The `Archive` utility does not delete the UOWs.

Upon completion, the `Archive` utility creates an XML file and a text file in the running directory. This XML file is used for restoring the archive. The XML file is an XML representation of the UOW table and related tables, and is used in archive restore if required to enter the data in again.

The text file produced is a report that lists total number of UOW archived and the original criteria for archiving.

Parameters

`Archive`
Specifies the keyword that instructs the `icosutil` utility to execute the `Archive` utility. The `Archive` utility proceeds to archive UOWs based on criteria specified in the `workArchivingUtility.properties` file.

`-f`
Specifies an option when using the `Archive` keyword.

`workArchivingUtility.properties`
Specifies the `workArchivingUtility.properties` file that defines the criteria that the `Archive` utility uses to archive UOWs from ITNCM - Base network resources.

Notes

As increasing numbers of network resources and configurations per resource are added to ITNCM - Base, housekeeping duties become particularly important to free up storage space and maximize run time performance. The following utilities perform these housekeeping duties:

Fix Pack 3

ConfigArchive
Deletes versioned configurations from the database.

For more information, see “Archiving and deleting versioned configurations from the database (ConfigArchive)” on page 120.

WorkHousekeeping
Deletes completed UOWs from the database.

For more information, see “Deleting completed UOWs from the database (WorkHousekeeping)” on page 123.

Archive
Archives the UOW.

Note: To perform Archive creation or restoring, the user must belong to a group with the Manage Archive activity.

Restoring a UOW from an archive (Restore)

To restore a UOW from an archive, specify the `Restore` keyword when executing the `icosutil` utility. This keyword executes the `Restore` utility with the options specified on the command line. One of these options is a properties file that allows you to specify which UOWs to restore from an archive.

Syntax

```
./icosutil Restore -f /opt/IBM/tivoli/netcool/ncm/config/properties/
workArchivingUtility.properties -archiveName xmlfilename
```

Description

The `Restore` utility restores the specified UOW and UOW logs from an archive. The `Restore` utility pulls the required information from the XML file specified in *xmlfilename* and restores the UOW based on criteria specified in the `workArchivingUtility.properties` file. By default,

the workArchivingUtility.properties file resides in the /opt/IBM/tivoli/netcool/ncm/config/properties directory.

Parameters

Restore
Specifies the keyword that instructs the icosutil utility to execute the Restore utility. The Restore utility proceeds to restore the specified UOWs based on criteria specified in the workArchivingUtility.properties file.

-f
Specifies an option when using the Restore keyword.

workArchivingUtility.properties
Specifies the workArchivingUtility.properties file that defines the criteria that the Restore utility uses to restore UOWs from an archive.

-archiveName
Specifies the name of the archive that contains the UOW that you want to restore.

xmlfilename
Specifies the name of the XML file that the Restore utility uses to pull the required information.

Clearing an archive (ArchiveDelete)

To clear an archive, specify the ArchiveDelete keyword when executing the icosutil utility. This keyword executes the ArchiveDelete utility with the options specified on the command line. One of these options is a properties file that allows you to specify which UOWs to clear from an archive.

Syntax

```
./icosutil ArchiveDelete -f /opt/IBM/tivoli/netcool/ncm/config/properties/
workArchivingUtility.properties -archiveName xmlfilename
```

Description

The ArchiveDelete utility clears the specified UOW and UOW logs from an archive. The ArchiveDelete utility clears the required information from the XML file specified in *xmlfilename* and clears the UOW based on criteria specified in the workArchivingUtility.properties file. By default, the workArchivingUtility.properties file resides in the /opt/IBM/tivoli/netcool/ncm/config/properties directory.

Parameters

ArchiveDelete
Specifies the keyword that instructs the icosutil utility to execute the ArchiveDelete utility. The ArchiveDelete utility proceeds to clear the specified UOWs based on criteria specified in the workArchivingUtility.properties file.

-f
Specifies an option when using the ArchiveDelete keyword.

workArchivingUtility.properties
Specifies the workArchivingUtility.properties file that defines the criteria that the ArchiveDelete utility uses to clear an archive.

-archiveName
Specifies the name of the archive that contains the UOW that you want to clear.

xmlfilename
Specifies the name of the XML file that the ArchiveDelete utility uses to clear the required information.

Chapter 7. Security

Use this information about Netcool Configuration Manager to understand system security.

TACACS+ authentication

ITNCM - Base has an external authentication capability to pass the username and password supplied at login GUI (or through the API) to an external custom class for authentication. The external authentication mechanism used is TACACS+.

TACACS+ is a remote authentication protocol, which allows a remote access server to communicate with an authentication server to validate user access onto the network. TACACS+ allows a client to accept a username and password, and pass a query to a TACACS+ authentication server. Login to ITNCM - Base is authenticated using the TACACS+ server instead of authentication locally.

There are significant benefits to be achieved from the implementation of external authentication:

- Improved Security — login authentication is more secure, as the ITNCM - Base user passwords are not held on a local database, instead it is managed and stored on a remote machine.
- Central Storage of Passwords — leverage existing password checking infrastructure. No need to duplicate account.
- Password Ageing — TACACS+ caters for password ageing, and notifies the user when the account has expired, and when it is about to expire.

Configuring the TACACS server

The TACACS server must first be configured on a different machine than the server running ITNCM - Base.

When a new user is created, or an existing user modified, there is an option to select "Remote User" within the user details. The Remote user checkbox when checked, describes the ability to use TACACS authentication to login for that particular user. When checked, both the Password and Validate Password fields are disabled, as this information must be configured within TACACS. The New User Screen has the following fields:

- *User Name:
- Remote User:
- *Password
- *Validate Password:
- First Name:
- Middle Initial:
- *Last Name:
- *E-mail:
- Telephone #:
- Identification:

Note: If the remote user option is left unticked, the password and validate passwords must be completed and the user will be saved locally.

Error messages

There are error messages associated with TACACS+ authentication.

Error Message	Description
This Password will soon expire. Please contact System Administrator for assistance.	The password set for the user on the TACACS+ Server is about to expire. This only works with ACSII authentication.
Either the user login password is expired/disabled, the login credentials are not correct or the TACACS+ Server is incorrectly configured. Please contact System Administrator for assistance.	The password set on TACACS+ Server has expired or been disabled, login details are incorrect or TACACS+ Server has been incorrectly configured.
There is a problem with the Host name. Check the auth.xml file to make sure the values are correct.	The host name of the TACACS+ server is incorrect, or has been incorrectly entered into the `auth.xml` file.
Problem found establishing a valid connection to the remote ACS Server. Either the values in the auth.xml file are incorrect or the TACACS+ server has not been configured correctly.	Incorrect values have been entered into the auth.xml file, and authentication fails. The `auth.xml` file must be checked to ensure that all details are correct. For example, if the `auth.xml` file contained details of a redundant TACACS+ server, authentication would fail.
There is a problem with authenticating TACACS+. Either the values in the auth.xml file are incorrect or the TACACS+ server has not been configured correctly.	Incorrect values have been entered into the auth.xml file, and authentication fails. The `auth.xml` file must be checked to ensure that all details are correct. For example, if a valid connection to the TACACS+ server is established, but the user has an incorrect secret key in their `auth.xml` file, authentication would fail.

AUTH.XML

The `auth.xml` file is configurable, and should be used to adjust settings for the TACACS server being used. For the purposes of TACACS authentication, the information within the `<tacacsPlus>` and `<backupTacacsServer>` XML tags, MUST be configured to modify TACACS server name, password, port number, client name, client port and authorization type.

Sample auth.xml file

The following example of an `auth.xml` file shows the required structure.

```
<tacacsPlus>
<name>TACACS Server name/IP Address</name>
<secret>Password</secret>
<port>Port number</port>
<client>Client Server name</client>
<clientPort>Client port number</clientPort>
<authType>Authorization Type, for example, ASCII</authType>
</tacacsPlus>
<backupTacacsServer>
<backupName>TACACS Server name/IP Address</backupName>
<backupSecret>Password</backupSecret>
<backupPort>Port number</backupPort>
<backupClient>Client Server name</backupClient>
<backupClientPort>Client port number</backupClientPort>
<backupAuthType>Authorization Type, for example, ASCII</backupAuthType>
</backupTacacsServer>
...
<protocolorder>
           <radius/>
```

```
            <intelliden/>-->
            <tacacsPlus/>
        </protocolorder>
```

Note: The <authType> XML tag supports all of the leading authentication protocols: ASCII, PAP, CHAP, ARAP, and MSCHAP.

Note: The <protocolorder> XML tag is used to list the order that the authentication types should be tried. The three options are: radius, intelliden and tacacsPlus.

Configuring Netcool Configuration Manager to use Active Directory authentication

Netcool Configuration Manager has an Active Directory authentication capability. Effectively, this works by using the LDAP settings in the existing company setup.

Login to Netcool Configuration Manager is authenticated using the Active Directory settings instead of authentication locally. There are significant benefits to be achieved from the implementation of Active Directory authentication:

- Improved Security - login authentication is more secure, as the Netcool Configuration Manager user passwords are not held on a local database, instead it is managed and stored on a remote machine.
- Central Storage of Passwords - leverage existing password checking infrastructure. No need to duplicate account.

There are two stages in the following process to configure Netcool Configuration Manager to use Active Directory for authentication. The first stage on the Active Directory platform involves creating users and groups organization units. The second stage is the configuration of Netcool Configuration Manager. Before proceeding, however, ensure you have backed up the original WebSphere security configuration.

Creating organization units

This task describes how to create users and groups organization units (OUs).

Before proceeding, please ensure the Netcool Configuration Manager server is running, and that you have backed up the original WebSphere security configuration.

Netcool Configuration Manager resources that are deployed within WebSphere are protected by specifying security roles in the Netcool Configuration Manager application descriptor files. Netcool Configuration Manager also installs a role-to-groups mapping within WebSphere. The following two groups are mapped to roles:

- IntellidenUser
- IntellidenAdminUser

During a user login Netcool Configuration Manager custom authenticator informs WebSphere which of these groups a user belongs to. When Active Directory is used as an authenticator, Active Directory must inform WebSphere of the groups that a user belongs to. To achieve this, both groups must be created in Active Directory and users must be added to the groups as follows:

- For a user to have access to all resources including the Account Management application, the user must belong to the IntellidenUser and IntellidenAdminUser groups.
- For a user to have access to all resources except the Account Management application, the user must belong to the IntellidenUser group.

If the Active Directory administrator is not permitted to create groups named IntellidenUser and IntelldenAdminUser, then names that are permitted must be used to create the groups, for example:

- SEC-Group ITNCM User
- SEC-Group ITNCM Admin User

In this case, the groups must be mapped to Netcool Configuration Manager roles by using WebSphere's wsadmin utility. Active Directory users must be added to the groups as previously described.

Note: There is no relationship between the groups created in Active Directory and groups created by the Netcool Configuration Manager Account Management application. The former are used by WebSphere to control access to Netcool Configuration Manager resources, whereas the latter are used by Netcool Configuration Manager to enforce security within its applications.

Important: You must create users in both Active Directory and the Netcool Configuration Manager database.

The example values used in the procedure are to be used as a guide. The values assume that an example domain named `itncm.local` has been created in Active Directory

1. Create a groups OU, for example `itncmgroups`, under `itncm.local`
2. Under this OU, create the following groups (or create two groups whose names are permitted):
   - IntellidenUser
   - IntellidenAdminUser
3. Create a users OU, for example `itncmusers`, under `itncm.local`
4. Under this OU, create the `Intelliden` user. The password you provide will be used in later steps.
5. Add the `Intelliden` user to the two groups you created in step 2.

Configuring Netcool Configuration Manager

This task describes how to configure Netcool Configuration Manager to use Active Directory authentication.

1. Launch the WebSphere Administrative Console:
   `http://<ncmserver-hostname-ip>:18100/ibm/console`
2. Log in using the Netcool Configuration Manager superuser name ('Intelliden') and password that was specified during installation.
3. Click **Security** > **Global security**
4. In the User account repository section, select **Federated repositories** from the Available realm definitions menu, and click **Configure**.
5. Enter a name in the Realm Name field.

   Note: You can use the default value (`defaultWIMFileBasedRealm`).
6. Enter `Intelliden` in the Primary administrative user name field, and click **Apply**.
7. Enter the Intelliden user password created in step 4 of “Creating organization units” on page 131 in the Password and Confirm password fields, click **OK**, and then click **Save directly to the master configuration**.
8. Select **Manage Repositories** and click **Add**.
9. Select **Microsoft Windows Active Directory** from the Directory type menu, and define the following details:

   Repository identifier
   Enter a value, for example AD

   Primary host name
   Enter the host name of the Active Directory server.

   Port
   Enter the port number of the Active Directory server.

   Bind distinguished name
   Enter the bind distinguished name, for example:

   ```
   cn=Intelliden,ou=itncmusers,dc=itncm,dc=local
   ```

   Bind password
   Enter the bind password. If the Intelliden user is the bind user, use the password created in step 4 of “Creating organization units” on page 131.

10. Click **Apply**, then click **Save directly to the master configuration**.
11. Click **Security** > **Global security**.
12. In the User account repository section, select **Federated repositories** from the Available realm definitions menu, and click **Configure**.
13. Click **Add Base entry to Realm**, and then select the **Active Directory repository identifier** from the Repository menu.
14. Define the following distinguished names:

    Base entry that uniquely identifies this set of entries in the realm field
    Enter the distinguished name, for example:

    ```
    dc=itncm,dc=local
    ```

    Base entry in this repository field
    Enter the distinguished name, for example:

    ```
    dc=itncm,dc=local
    ```

15. Click **Apply**, then click **Save directly to the master configuration**.
16. Click **Security** > **Global security**.
17. In the User account repository section, select **Federated repositories** from the Available realm definitions menu, and click **Configure**.
18. In the Repositories in the realm table, select the repository whose identifier is InternalFileRepository.
19. Click **Remove**, then click **Save directly to the master configuration**.
20. Click **Security** > **Global security**.
21. In the User account repository section, select **Federated repositories** from the Available realm definitions menu, and click **Set as current**.
22. Click **Apply**, then click **Save directly to the master configuration**.
23. If the Active Directory groups were named `IntellidenUser` and `IntellidenAdminUser`, go to step 24, otherwise proceed to "Configuring Netcool Configuration Manager roles" on page 133.
24. Log out of the WebSphere Administrative Console, and restart Netcool Configuration Manager: `<itncm_install_dir>/bin/./itncm.sh restart`

    Note: Use the Intelliden user password that was specified during installation.

Once Netcool Configuration Manager has restarted, the Intelliden user password will become what you provided for the Intelliden user in Active Directory.

Remember: Existing Netcool Configuration Manager users must be created in Active Directory. New users must be created in both Netcool Configuration Manager and Active Directory. In all cases the user password is the password that is provided in Active Directory.

Note: If the user tries to restore the previous configuration of WebSphere while using AD authentication, it is advisable to take system backup. The backup can be used if in case configuration restore action corrupts the deployment. Refer to WebSphere documentation on how to disable Global Security, revert to the Standalone custom registry, then re-enable Global Security before taking backup.

When backing up WebSphere, it is strongly recommended not to use `backupConfig.sh` or `restoreConfig.sh` scripts for performing backup and restore procedure on Jazz. For more information, refer to Can JazzSM server be backup and restore using these scripts-backupConfig.sh and restoreConfig.sh? tech note.

Configuring Netcool Configuration Manager roles

Use this procedure if the Active Directory groups are not named `IntellidenUser` and `IntellidenAdminUser`.

Before proceeding, please ensure the Netcool Configuration Manager server is running.

The group names used in the procedure, `SEC-Group ITNCM User` and `SEC-Group ITNCM Admin User`, are examples and must be replaced with the actual values. Replace all occurrences of <password> in the procedure with the Intelliden user password. If Active Directory is being used to authenticate the user, use the password that is stored in Active directory, otherwise use the password that was specified during the Netcool Configuration Manager installation.

1. Open a command line terminal on the Netcool Configuration Manager Presentation server:
   `cd <itncm_install_dir>/eWAS/bin`
   a) Issue the following command:
   ```
   ./wsadmin.sh -connType SOAP -user Intelliden -password <password> -c
   '$AdminApp edit "Intelliden R-Series" {-MapRolesToUsers {{"IntellidenUser"
   no no "" "SEC-Group ITNCM User" }} }'
   ```
   b) Issue the following command:
   ```
   ./wsadmin.sh -connType SOAP -user Intelliden -password <password>
   -c '$AdminApp edit "Intelliden R-Series" {-MapRolesToUsers
   {{"IntellidenAdminUser" no no "" "SEC-Group ITNCM Admin User" }} }'
   ```
   c) Issue the following command:
   ```
   ./wsadmin.sh -connType SOAP -user Intelliden -password <password> -c
   '$AdminApp edit "Intelliden R-Series" {-MapRolesToUsers {{"Intelliden" no
   no "" "SEC-Group ITNCM User" }} }'
   ```
   d) Issue the following command:
   ```
   ./wsadmin.sh -connType SOAP -user Intelliden -password <password> -c
   '$AdminApp edit "PBCM" {-MapRolesToUsers {{"IntellidenUser" no no "" "SEC-
   Group ITNCM User" }} }'
   ```
   e) Log out of the WebSphere Administrative Console, and restart Netcool Configuration Manager:
   `<itncm_install_dir>/bin/./itncm.sh restart`
2. Launch the WebSphere Administrative Console:
   `http://<ncmserver-hostname-ip>:18100/ibm/console`
3. Configure the non-administrative user group:
   a) Click **Environment** > **Naming** > **Name Space Bindings**
   b) In the **Scope** section, select **Node=JazzSMNode01, Server=server1**, unless your administrator tells you to use a different server.
   c) Click **New...** > **String**.
   d) Click **Next**.
   e) Type `Intelliden User Group Name` as the Binding identifier.
   f) Type `ncm/group/IntellidenUser` as the name.
   g) Type the name of the LDAP group that controls user access in the **String value** field.
   This group is the custom equivalent of the `IntellidenUser` default group name.
   h) Click **Next**, **Finish**, and save the configuration change.
4. Configure the administrative user group:
   a) Click **Environment** > **Naming** > **Name Space Bindings**
   b) In the **Scope** section, select **Node=JazzSMNode01, Server=server1**, unless your administrator tells you to use a different server.
   c) Click **New...** > **String**.
   d) Click **Next**.
   e) Type `Intelliden Admin User Group Name` as the Binding identifier.
   f) Type `ncm/group/IntellidenAdminUser` as the name.
   g) Type the name of the LDAP administrative group that controls user access in the **String value** field.
   This group is the custom equivalent of the `IntellidenAdminUser` default group name.
   h) Click **Next**, **Finish**, and save the configuration change.

5. Restart the Netcool Configuration Manager server.
6. Close and reopen the browser.
7. Log in again.

Netcool Configuration Manager - Compliance security

Access management, device security, levels of user permission and security administration are explained. Instructions are also provided for the configuration of all functionality.

User authentication

In Netcool Configuration Manager - Compliance management the term authentication refers to the means by which the system positively identifies each user. The compliance security subcomponent leverages the authentication framework used in the main Netcool Configuration Manager security component to authenticate users.

Device security

Netcool Configuration Manager - Compliance functionality leverages the resource (realm, device) access that can be set up in Netcool Configuration Manager.

Scope

A user's data scope is defined in terms of the group's realms and sub-realms as defined in Netcool Configuration Manager.

Remember: Realms are predefined hierarchies of network resources organized physically or logically.

Network resources defined in a particular realm that is within a group's data scope are accessible by that groups' users. For further information about the use of realms, see the *IBM Tivoli Netcool Configuration Manager User Guide*.

Security

The ability to act on particular realms and the contents of the realms is controlled using the security tab within the Netcool Configuration Manager Account Management **Group** window. Users' rights to realm and resources within these realms are inherited from any group to which they belong.

User privileges for device security within the Netcool Configuration Manager - Compliance UI are firstly applied in Netcool Configuration Manager. The privileges may be applied so that users may only select devices on the network to which they have the appropriate access as per device access defined in the main Netcool Configuration Manager security component.

Compliance Validation Users can therefore only view device realms and devices to which they have also been granted access in the main Netcool Configuration Manager security component.

Additional group permissions

The groups listed here are reflective of the Netcool Configuration Manager Account Management group account administration area. This is a convenient method for checking group-user membership.

1. In the Netcool Configuration Manager - Compliance UI, click **Admin** > **User Security Options** on the menu bar.

   The **Netcool Configuration Manager - Compliance Security Administration** window is displayed.
2. Optional: On the **Group Permissions** tab, select a group and then select the **Show Users** button to display a list of user IDs with membership of that group.
3. On the **Group Permissions** tab, select a group and then select the **Group Options** button.

   The **Options for group** dialog is displayed.
4. Select one or more of the following group options:

*Table 8. **Netcool Configuration Manager - Compliance Security Administration** window, Group Permissions tab, Group Options dialog options*		
Option	**Type**	**Description**
Use login credentials when submitting interrogation	Checkbox	When executing a policy that contains a native command, it submits an interrogation UOW. When unchecked, the default auto-approval user's credentials are used to submit the interrogation UOW. When checked, the current user credentials are used. For information about configuring the alias of the auto-approval user, see *Change Compliance user names and passwords using the intellidenRmUser.sh script*.
Use login credentials when submitting remedial action	Checkbox	In the event that a remedial action UOW is submitted. When unchecked, the default remedial action user's credentials are used to submit the remedial action UOW. When checked, the current user credentials are used.
Automatic submission of remedial actions	Checkbox	When checked, this setting enables automatic approval of remedial actions without the need for manual intervention in the Queue Manager Results pane.
Device login credentials override	Drop-down menu	The Override Credentials step will appear in the creation of a process or in the execution of a policy only if the user is a member of a group with this option set to either Optional or Required. **Hidden** The default device credentials will be used. **Optional** Default credentials, or own credentials supplied. **Required** Own credentials must be supplied.

5. Click the **Users** tab.

   Use the following table to understand the information displayed on the **Users** tab:

*Table 9. **Netcool Configuration Manager - Compliance Security Administration** window, Users tab*	
Field name	**Description**
Login	Full name or description attached to the user login
User ID	User ID used for authentication
Email	Email address of user

6. Click the **Realm Access Control** tab.

   Existing realms are listed along the left hand side of the tab, resembling folders. When you select a realm, the groups permitted to view that particular realm are displayed in the 'Allowed Groups' section on the right hand side of the screen.

7. Perform one or more of the following actions:
   - Add or remove groups from any realm.
   - Create new realms.
   - Edit existing realms.
   - Delete realms.

Change Compliance user names and passwords using the intellidenRmUser.sh script

Use the CLI to change user names and passwords.

Using the script

Use the `/opt/IBM/tivoli/netcool/ncm/compliance/bin/utils/intellidenRmUser.sh` script to change the user ID and password of Netcool Configuration Manager - Compliance users. All user names and passwords are encrypted on Netcool Configuration Manager - Compliance.

You might want to use this script as part of changing Netcool Configuration Manager - Compliance group permisisons. For more information, see *Additional group permissions*.

To retrieve the user ID and password of a Netcool Configuration Manager - Compliance user, navigate to the `/opt/IBM/tivoli/netcool/ncm/compliance/bin/utils/` directory and issue a command similar to the following:

```
./intellidenRmUser.sh --get user
```

where *user* is one of the following:

- `cmuser` - the automatic approval user
- `rmuser` - the remedial user
- `automateduser` - the automated processes user

To set the user ID and password of a Netcool Configuration Manager - Compliance user, navigate to the `/opt/IBM/tivoli/netcool/ncm/compliance/bin/utils/` directory and issue a command similar to the following:

```
./intellidenRmUser.sh --set user new-user-ID new-password
```

where *user* is one of `cmuser`, `rmuser`, or `automateduser`; *new-user-ID* is the new user ID; and *new-password* is the new password.

Related information

User administration
Use this information to administer Netcool Configuration Manager users.

Insufficient security

Within the ITNCM-Compliance application, users are bound by ITNCM - Base based permissions, such as the corrective actions they can trigger from the compliance violation queue.

As a requirement, ITNCM-Compliance needs to have at least one user with full approval rights. This means they must have access to the "Manage Work" activity in the ITNCM - Base account administration (this activity allows a UOW to be processed automatically). The reason for this is auto-approval of command sets, and the procedure for sending show commands. It is deemed acceptable for Remedial command set work to be queued up, however show command work may not be.

There are problems associated with having insufficient security to execute remedial command sets. If a user does not have access to the "execute direct commands" activity in the ITNCM - Base account administration (this activity is used for running command sets), ITNCM-Compliance shall pass the command set into ITNCM - Base, where it will sit in the approval queue as a UOW until someone who has the appropriate access approves it. The command set will receive a corresponding UOW ID in ITNCMCompliance, which will be used to identify it, and used also to listen for the UOW to return.

See the *ITNCM User Guide* for more information about the functioning of the approval queue.

Chapter 8. OS Manager

Use this information about Netcool Configuration Manager to understand the OS Manager.

About OS Manager

OS Manager is a powerful tool that has the ability to update hundreds of network devices simultaneously. The operating systems of these network resources can be easily upgraded using the OS Manager user interface. Individual resources may be specified for an upgrade, or all resources of a given VTMOS may be updated.

The Network Specialist is responsible for creating and maintaining the OS Registry in ITNCM - Base. Typically, a new OS would be downloaded from the hardware vendor's support web site. For devices not currently on the ITNCM - Base OS Upgrade supported list, new support can be added by creating a new OS upgrade device script. The OS Upgrade utility allows the user to define their own “user-defined” parsers using Regex. The user must ensure they only include supported operating systems in the OS Registry XML file.

When a new OS has been applied against a network resource, the user not only has to upgrade to the new OS, but they must also execute a configuration change and reboot the resource in order to move the new OS to “running”. A config sync is then also required to get the database copy of the configuration in sync with the running and current configs on the resource. The OS Upgrade utility includes options to perform these other necessary steps.

Security rights

In order to update the OS on a network resource, any user must belong to an Netcool Configuration Manager - Base security group with the following permissions:

- OS Upgrade
- Execute Direct Commands
- Manage Work
- Execute Configuration Synchronization activities

The groups must also have view and modify rights on the appropriate realms.

High-level procedure

The following steps provide the high-level procedure required to upgrade the OS on one or more network resources. The remainder of this chapter describes the specifics of each step.

1. Create FTP Resource. See the Netcool Configuration Manager - Base User Guide for further information.
2. Create an OS Registry.
3. Create an OS Specification Resource.
4. (Optional) Create a command set to be run during the Upgrade.
5. Select the Resources to upgrade.
6. Schedule the OS Upgrade UOW.

OS registry

The OS registry is a database of compatible operating system image files that have been approved by a qualified network specialist for upgrade on a particular VTMOS. The OS Registry may be added to, removed from, and edited.

Note: Best practice for Juniper upgrade: Juniper recommends the target file system be `/var/tmp`. Problems may occur if this practice is not adopted. For example, if TACACS+ is used for accessing the device, and the user is neither a Superuser nor in a group, the user directory will fill up with data making it difficult to manage free space on the flash system. The group must be assigned most, if not all rights. If there is no group then the user must be provided with read/write permissions.

Creating an OS registry

Create a new OS Registry by filling in the fields presented by the New OS Registry dialog.

Before creating a new OS Registry, ensure that a qualified network specialist has approved the associated operating system image files for upgrade on a particular VTMOS.

This task creates a new OS Registry.

1. From the ITNCM - Base Resource Browser, select **File** > **New OS Registry**. Alternatively, right-click within the Resource Browser, and choose **New** > **OS Registry**.
2. The New OS Registry dialog is displayed.
   a) Specify an OS Registry name, for example, Cisco OS Reg.
   b) Specify the vendor, for example, Cisco.
   c) Specify the type of the OS Registry, for example, Firewall.
   d) Specify the Model, for example, 5*.
   e) Specify the operating system.
3. Click **OK** to create the specified OS Registry. The OS Registry will be created in the Resource Browser.

To configure the OS Registry, you must edit it.

Editing an OS registry

Configure an OS Registry by using the Edit OS Registry tabbed dialog.

You must have previously created the OS Registry by filling in the menu items displayed on the New OS Registry dialog.

To edit (configure) an OS Registry, follow these steps.

1. Highlight the OS Registry resource, and select File | Edit from the toolbar. Alternatively you may right click within the Resource Browser, and choose Edit.
2. The Edit OS Registry tabbed dialog is displayed. Use this dialog to configure and maintain the OS Registry. The dialog has two tabs: **OS Registry** and **Edit in XML Format**. In the OS Registry tab, you have the ability to add/remove file system types, connect to an FTP Server to download new OS images, and the ability to add/update/remove OS. Use the following table to understand the items available with the Edit OS Registry dialog:

Option	Description
Dialog item	Description
File System Type	Not all devices make use of flash, disk or slot, but will use other types of data storage styles. For example, Juniper makes use of a Unix based file structure, which would look like `/var/tmp`. **Important:** If the image file is in a subdirectory on the device, you must create a File System Type with the full path of the subdirectory. For example: `flash:/newdir`

Option	Description
OS Name	Identifies the version of OS. This is what the user will see when building the OS Specification and the wizard prompts for the OS version.
OS Image	This is the file name including extension type. If this file is named incorrectly, or does not match a file on the ftp/tftp server, the UOW will fail with a message saying the file could not be found.
Memory (MB)	Minimal amount of memory needed to run the OS on the device. This information is provided from the OS Vendor. If this value is lower than the actual available memory on the device identified by the parser, the UOW will fail due to not enough memory.
Image Size	Actual size of the OS file being copied to the device. Size can be obtained by issuing a `ls -d` or `dir` depending on the OS hosting the ftp/tftp server. If this value is higher than the value identified via the hardware parser, the OS upgrade will fail due to not enough memory.
Comment	Narrative entered here will be seen in the OS Specification when the user chooses the OS required.

3. If you decide to connect to an FTP Server, the FTP Connection screen is displayed. Select an FTP resource by clicking on one of the FTP resource entries displayed in **Select FTP Device** and then click the **Load** button. The details for the FTP resource display in **FTP Server Details**. This screen gives you the ability to choose an FTP resource, and load the FTP Server details into the screen. Use the Connect button to provide login credentials, establish a connection to the FTP Server, and retrieve a listing of OS images from an FTP server. Use the following table to understand the items available with the FTP Connection screen:

Option	Description
Screen Item	Description
Server Name:	Displays the name of the selected FTP server. For example, `10.216.1.171`.
User Name:	Displays the name of the user who has access to the selected FTP server.
User Password:	Displays the encrypted password for the user who has access to the selected FTP server.
FTP Server Path:	Displays the path to the selected FTP server.

4. The existing OS images in the OS Registry may also be updated from the OS Registry tab, if the Update button is chosen. The details on this screen may be amended if necessary. Note that if the RAM requirements are not known, this should be checked on the vendors website.
5. When maintenance on the OS Registry is complete, click **Save** or **Exit**.
6. If you wish to edit the XML directly, click the **Edit in XML Format** tab. You can manually add a list of OS Image files in the XML Format tab, using the `<image>`, `</image>` tags. This is helpful if you want to bypass the connection to the ftp server to retrieve a list of images. This same editing window is used for editing security sets, access properties, authentication resources, OS Registries, and so forth.
7. If you have made changes, click **Save**.

You can now create an OS specification resource.

OS specification

An OS specification provides details about a device's operating system.

You create an OS specification by following the steps that the **Select VTMOS** dialog provides. You edit an OS specification by using the **Edit OS Registry** tabbed dialog.

Creating an OS specification

Create an OS specification by using the Select VTMOS screen.

To create a new OS specification, follow these steps.

1. From the ITNCM - Base Resource Browser, select **FileNew** > **OS Specification**. Alternatively, right-click within the Resource Browser, and choose **New** > **OS Specification**. The Select VTMOS screen is displayed. Note that the **1. Select VTMOS** item is highlighted. Supply a name for the OS specification resource and the VTMOS for the device to be updated. Then click Next to continue. The following table describes each of the fields in the screen.

Option	**Description**
Screen item	**Description**
Name:	Specifies the name for this OS specification resource. For example, `NewOS_Spec`.
Vendor:	Specifies the vendor that corresponds to the vendor of the device to be updated. For example, `Cisco`.
Type:	Specifies the type that corresponds to the device to be updated. For example, `Firewall`.
Model:	Specifies the model that corresponds to the device to be updated. For example, 5*.
OS:	Specifies the operating system that corresponds to the device to be updated. For example, *.

2. The Select OS Registry screen is displayed. Note that the **2. Select VTMOS** item is highlighted. Select the correct OS Registry for use with the upgrade. Then click Next to continue.
3. The OS Version to Upgrade to screen is displayed. Note that the **3. Select OS Version to Upgrade To** item is highlighted. From the Target OS drop down menu, select the version to upgrade to. When the selection has been made, this will populate the OS Details information into the lower section of the screen. Click Next to continue.
4. Because many network resources lack the disk space needed to load a new OS, the utility provides the opportunity to specify files to delete in order to free up space. The Select Files to be Removed screen is displayed. Note that the **4. Select Files to be Removed** item is highlighted. To specify the files to delete:
   a) In **Image Destination**, select the target file system.

      Note: If the image file is in a subdirectory on the device, you will have created a File System Type with the full path of the subdirectory previously. You select that file system.

      

      CAUTION: The **Erase All** checkbox option when checked will erase the entire file system using the download section of the device script.

   b) To erase specific file systems, remove the check mark in the **Erase All** checkbox. In this case, the copydown section of the device script is used to erase the specified file system. If updating the device with the same OS image, the original image will automatically be removed and replaced by the selected target image.

   c) The **Make Room On Destination** option allows you to make additional room on the destination. In order to upload a new OS image to a network resource, it is sometimes necessary to clear up disk

space. You can identify specific files to delete. Or, you can specify a wildcard (for example, `*.bin`) that recursively removes all files with a `.bin` extension.

d) Click **Next** to continue.

5. (Optional) The Select Boot Command Set screen is displayed. Note that the **5. Select Boot Command Set** item is highlighted. It may be necessary to create a modelled command set to update the value of the configuration register on the device. Subsequently, a user can select this Command Set to modify boot parameters prior to reloading the new OS. Click Next to continue.
6. The Select Parser screen is displayed. Note that the **6. Select Parser** item is highlighted. Specify the parser to use for file system and memory checks. There are two choices: Device Content Parser (Default) and User Defined Parser (Advanced). The Device Content Parser is the ITNCM - Base parser installed with a driver. Alternatively, you can specify a user defined parser for unique cases where the ITNCM - Base parser does not work. The user defined parser can be manipulated using Regex.
7. If you choose the User Defined Parser, the User Defined Parser screen is activated. Note that the **7. User Defined Parser** item is highlighted. The User Defined Parser screen shows six keys for which a value must be returned. The keys for the OS Upgrade are: `findcurrentOsImageName`, `findMemoryTotal`, `findfilesystemMemoryTotal`, `findSystemMemoryFree`, `findfilelist`, and `findfilesize`. Each key of information can be retrieved by performing the relevant `show` command on a device. For example, the `findMemoryTotal` value can be returned on a CISCO device using the `show version` command and using this regular expression to extract the value `'(\d+)K\/\d+K bytes of memory'`. After updating any keys displayed in the User Defined Parser screen, click Next to continue.
8. The Describe Work screen is displayed. Note that the **8. Describe Work** item is highlighted. Use this screen to describe the OS Specification Resource, or to provide any comments.

The following is an example of the download section of the device script (used when erasing an entire file system).

```
downLoad.01.send=copy tftp $copy_input2$\r
```

The following is an example of the copydown section of the device script (used when erasing a specific file system).

```
copyDown.01.send=copy ftp://
$ftp_altusername$:$ftp_altpassword$@$ftp_althostname$/
$ftp_altpath$/$copy_input1$ $copy_input2$\r
```

To edit an OS specification, use the Edit OS Registry tabbed dialog.

Editing an OS specification

Edit an OS specification by using the Edit OS Registry tabbed dialog.

To edit an OS specification, follow these steps.

1. Highlight the OS Specification resource, and select **File** > **Edit** from the toolbar. Alternatively, you may right click within the Resource Browser, and choose **Edit**.
2. The **Edit OS Registry** tabbed dialog is displayed. This Edit OS Registry dialog replicates all the screens configured in the creation of an OS specification, and allows you to make amendments to the previous choices made when creating the OS specification. Note that when performing the actual OS Upgrade, ITNCM - Base uses the information entered into the OS Registry resource to decide if there is enough room for the new image. It is important to correctly specify the size and memory requirements of the new OS image. When you have completed all edits to the OS specification, click **Save**. A message is displayed indicating that the OS Specification resource has been saved.

You still need to submit and schedule the OS upgrade. For more information, see “Submitting an OS upgrade request” on page 145.

Creating an OS upgrade device script

ITNCM uses the device scripts in the appropriate RAD for OS Upgrades. But if you need separate device scripts for OS Manager you can create them by following this section.

To create an OS Upgrade device script, follow the steps below, first ensuring that any RAD you have explicitly created for the device has Access Types / Command Line / Script set to the default value rather than a concrete value such as `deviceScript-default` or `deviceScript-ssh`.

1. Right click in the Resource Browser and select **New** > **OS Upgrade** device script. The New OS Device Script screen is displayed.
2. Specify the name and VTMOS of the new OS upgrade device script and click **OK**. This action creates the OS device upgrade script in the Resource Browser under the current realm. The following table describes and provides examples for each of the fields in the screen.

Option	**Description**
Screen item	**Description**
Name:	Specifies the name for this OS upgrade device script. For example, `Cisco OS REG`.
Vendor:	Specifies the vendor that corresponds to the vendor of the device for which you are supplying an OS upgrade device script. For example, `Cisco`.
Type:	Specifies the type that corresponds to the device for which you are supplying an OS upgrade device script. For example, `Firewall`.
Model:	Specifies the model that corresponds to the device for which you are supplying an OS upgrade device script. For example, `5*`.
OS:	Specifies the operating system that corresponds to the device for which you are supplying an OS upgrade device script. For example, `*`.

3. You can edit the newly created OS upgrade device script at any time by right clicking on the resource and selecting **Edit**.
4. The sections of the device script that the OS Manager tool uses uniquely are: *del*, *squeeze directory*, *erase*, *copydown*, *download*, and *reload*. Each of these sections has variables to pass down the relevant arguments to the device CLI during execution of the upgrade. The following table describes each of the relevant variables:

Option	**Description**
Relevant variable	**Description**
del_input	Specifies the filename to be deleted on the device. Populated according to user preference in the OS Specification, that is, dynamically by specifying wildcard `*.bin` to remove all bin files or by specifying individual files.
$squeeze_input$	Contains the file system name to be squeezed, for example, flash. Populated according to the file system defined by the user in the OS Specification.
$ftp_altusername$	Specifies the username to connect to an FTP server. Populated from the file transfer resource section altFtpInfo.
$ftp_altpassword$	Specifies the password to connect to an FTP server. Populated from the file transfer resource section altFtpInfo.
$ftp_althostname$	Specifies the hostname of the FTP server to connect to. Populated from the file transfer resource section altFtpInfo.

Option	Description
$copy_input1$	Contains the name of the file to transfer from the TFTP/FTP server. This variable is used in the *copydown* section and is populated from the selection that a user makes in the OS Specification.
$copy_input2$	Contains the name of the file system on the device to which the OS image file will be downloaded, for example, flash or boot flash. The user specifies the name of the file system in the OS Specification.
$erase_input$	Specifies the name of the file system to be erased. This variable is used if the user has specified in the OS Specification to erase an entire file system.

5. If the image file is in a subdirectory on the device, you must replace colons with backslashes, because devices do not use colons in paths for subdirectories.
   For example, you change `$cd_input$:` to `$cd_input$/` wherever the file target is specified.

You submit an upgrade request by following the instructions in "Submitting an OS upgrade request" on page 145.

Submitting an OS upgrade request

There are different ways to submit an OS upgrade request. An OS Specification resource may be selected and applied to one or more network resources, or one or more network resources may be selected and an OS Specification resource then applied.

An OS Upgrade may also be applied to a realm. When applying to a realm, a VTMOS combination is specified and all resources in the realm with the specified VTMOS will have the new OS loaded. When applying to a realm, the user can also specify whether sub-realms should also be included.

Note: The `vsftpd` daemon that is installed by default on Red Hat systems will not work with OS Upgrades when using the FTP protocol. It is recommended to use the `proftpd` daemon.

To submit and OS upgrade request, follow these steps.

1. From the ITNCM - Base Resource Browser, choose the required network resources. Select Tools| OS Upgrade. Alternatively, you may right click within the Resource Browser, and choose Tools | OS Upgrade.
2. The Select OS Specification dialog is displayed. Note that the **1. Select OS Specification** item is highlighted. Choose the OS Specification resource to be applied, and then click Next.
3. The Select the Scope of Application (Page 1 of 2) dialog is displayed. You need to choose whether the selected OS specification will be applied to specific network resources or to network resources within a realm. The three choices are: Apply OS Specification to network resources in a realm, Apply OS Specification to specific Network Resources, or Apply OS Specification to the Network Resources retrieved from a realm. Make the selection, and click Next.
4. The Select the Scope of Application (Page 2 of 2) dialog is displayed. Note that the **2. Select the Scope of Application** item is highlighted. Using the navigation tree in the Device Pane, select the necessary devices or realms that are required to submit an OS upgrade against. Then click Next.
5. The Configure Execution Options wizard page is displayed. Note that the **3. Configure Execution Options** item is highlighted.
   a) Use the following table as a guide to entering the appropriate information for the **Execution Mode** section of the Configure Execution Options wizard page.

Table 10. Execution Options

Exection Mode item	Description
Execute Mode (may possibly change all selected Network Resources)	Select this item to apply an OS Upgrade to selected Network Resources.

Table 10. Execution Options (continued)	
Exection Mode item	**Description**
Report Only Mode (does not change any Network Resources)	Select this item to produce a report that details the changes that would be made to those resources if in Execute Mode. This item does not change resources. The OS Upgrade will perform a check to confirm there is enough RAM and file system memory on the device for the new OS. The OS Upgrade also confirms if the image file exists on the FTP server.

b) Use the following table as a guide to entering the appropriate information for the **Failure Options** section of the Configure Execution Options wizard page.

Table 11. Failure Options	
Failure Options item	**Description**
Ignore All Errors	Indicates that the UOW will continue processing, regardless of any failures that occur. If each command set is to be applied to each resource regardless of any errors, select the Override flag as well.
Fail After X Total Errors	This item allows a user to select how many errors can occur before total failure of the process.
Fail After X Percent Errors	This item allows a user to select the maximum percentage of failures that can occur before the UOW stops processing.

c) Click Next to continue.

6. (Optional) The Password Override screen is displayed. Note that the **4. Password Override** item is highlighted. This is an optional step, in the event the user wishes to override the ITNCM - Base Authentication.
7. The Schedule Work screen is displayed. Note that the **5. Schedule Work** item is highlighted. Use the following table as a guide to entering the appropriate information for the **Schedule Work** section of the Schedule Work screen. Click Next when you are finished.

Option	**Description**
Schedule Work item	Description
Single Schedule	Select **Immediate** if you want an unscheduled process. Select **Scheduled** if you want to schedule a recurring execution of the process.
Scheduled Start	If you selected a recurring execution of the process, specify a time and date for when the execution of the process should start.
Scheduled End	If you selected a recurring execution of the process, specify a time and date for when the execution of the process should end.

8. The Execution Priority dialog is displayed. Note that the **6. Execution Priority** item is highlighted. By default, all UOWs are submitted with a priority of Medium. Use this dialog to change the priority from Medium to something more appropriate. Click Next to continue.

9. The Work Conflicts screen is displayed. This screen is displayed in the event that there are any conflicts in the UOW running against the chosen network resources. The user has the choice to Override Conflicts to continue with the submission, or they may Remove Resources to remove the conflicting resources. Click Next to continue.
10. The Select Workflow Options screen is displayed. Note that the **8. Select Workflow Options** item is highlighted. Use the following table as a guide to entering the appropriate information for the **Select Workflow Options** screen. Click Next when you are finished.

Option	Description
Workflow Option	Description
Synchronize device pre OS Upgrade	By default the device will be synchronized before the OS Upgrade is run.
Resource Check pre OS Upgrade	Select this item if you want to run checks on the RAM/flash memory.
Employ user defined file system and memory parsers	Select this item if you want to perform RAM/flash memory check using User Defined Parsing.
Delete file or erase the file system, then download the image to the device	Select this item if you want to delete files and then transfer the image.
Reload device	Select this item if you want to reboot the device after the OS upgrade has been performed.
Synchronize device post OS Upgrade	Select this item if you want to synchronize the device after the OS upgrade has been performed.

11. The Describe Work screen is displayed. Note that the **9. Describe Work** item is highlighted. Enter a description to identify the UOW, and click Finish to complete the OS upgrade submission request.

You can choose to model an OS manager.

Modeling OS manager per device

You can model an OS manager per device.

To model an OS manager per device follow these steps.

1. Visit the Vendor's web site and identify the device OS Upgrade commands that should be used, and determine best practices for upgrading. It is advised you should test the commands manually on the device to understand the OS upgrade process.
2. Update the device script of the device for which the support is needed with the appropriate commands. The sections of the device script that the OS Manager tool uses uniquely are: *del*, *squeeze directory*, *erase*, *copydown*, and *reload*. For information on device scripts, see “Creating an OS upgrade device script” on page 144.
3. Check whether the ITNCM - Base parser is already available for the device. Cross check that the system memory and file system memory is retrieved correctly.
4. If the parser does not exist, a user defined parser must be created in the OS Specification resource. For information on how to create the OS Specification, see “Creating an OS specification” on page 142.

Creating and editing an FTP Resource

You set up an FTP Resource to work with OS Manager.

Instead of FTP, you can define another transfer protocol to be used, such as TFTP or SCP.

1. In the Resource Browser, select the realm where you would like to create the FTP Resource. Typically, this is the same realm as the device that is to be upgraded.
2. Click **File** > **New** > **File Transfer**.

   The New File Transfer resource is displayed
3. Enter a `Name` for the new FTP resource, and provide applicable VTMOS data, as required.
4. Click **OK** to create the FTP Resource.

   The resource is created and displayed in the Resource Browser.
5. Highlight the FTP Resource, and select **File** > **Edit** from the toolbar. Alternatively, right-click within the Resource Browser, and choose **Edit**.
6. Click **Add** to create a default entry, and provide the following details:

   Name
   Enter `altFtpInfo`

   Host
   Enter the hostname of your FTP server.

   Username
   Enter the username of your FTP Server.

   Password
   Enter the password for your FTP Server username.

   Path
   Specify the path on the FTP Server where OS images are located relative to FTP user home directory.

   If OS images are in the FTP user home directory, enter `/`

   Passive mode
   If passive mode is required, select the check box.

   If not selected, Active Mode is used (this is the default).
7. Click **Save**.
8. If TFTP or SCP protocols are required instead of the default FTP, perform the following action:
   a) Highlight the FTP Resource, and select **File** > **Edit** from the toolbar.
   b) Select the **XML** option on the toolbar.
   c) Locate the uncommented `altFtpInfo` entry.
   d) For TFTP, add, `<type>tftp<type>`, or for SCP, add `<type>scp<type>`.
9. The FTP Resource supports using a fully qualified path to the OS images. In the XML, add the following xml to your `altFtpInfo` entry, and provide a full path to the OS images:
   `<fqpath>/home/icosftp<fqpath>`

Chapter 9. OOBC software

Use this information about Netcool Configuration Manager to administer the OOBC software.

Starting and stopping the OOBC daemon

Start and stop the OOBC daemon by using the `start` and `stop` arguments associated with the `oobc.sh` script.

For the Unix platform, there is the `oobc.sh` script which follows typical Unix daemon conventions in that it can be invoked with one of three arguments: start, stop, and restart.

This task explains how to start and stop the OOBC daemon by executing the `oobc.sh` script with the appropriate argument.

Change the path to the desired directory. There is no recommendation for this, it can be placed anywhere.

a) To start the OOBC daemon, execute the `oobc.sh` script and specify the `start` argument:

```
# ./oobc.sh start
```

b) To stop the OOBC daemon, execute the `oobc.sh` script and specify the `stop` argument:

```
# ./oobc.sh stop
```

Resetting the password in the oobc.properties.xml file

Reset the password in the `oobc.properties.xml` file whenever the Netcool Configuration Manager - Basepassword changes for the OOBC user.

The `oobc.properties.xml` file resides in the product installation directory. However, since the password is always stored in an encrypted format, you must use a command line utility to encrypt the password.

Note: This utility will not change the password within Netcool Configuration Manager - Base. It is only a mechanism to take a clear text password, the new password within Netcool Configuration Manager - Base, and encrypt it for use by the OOBC daemon.

This task explains how to reset the password in the `oobc.properties.xml` file.

1. Change directory to the product install directory:

```
/OutOfBandChange
```

2. Run the password encryption utility and enter the password that is current for the user within Netcool Configuration Manager - Base. For example:

```
# ./install.sh gen-pass
Enter clear text password:
secretPassword
Encrypted password is 6ff959c25fca02b600510808d86f852b
Edit the oobc.properties.xml file and update the password field.
BUILD SUCCESSFUL
Total time: 5 seconds
```

3. Edit the `oobc.properties.xml` file in the product install directory and replace the old <password> value with the newly encrypted password output from the password encryption utility.

OOBC Syslog files

Use this information to set the `syslogMessageSaverFile`, and to understand the rollover strategy for `syslog` files.

Description

You can specify a location to which the `syslog` files should be saved to. The `syslog` filepath is determined by the `syslogMessageSaverFile` parameter in the `oobc.properties.xml` property file. You should enter an appropriate filepath for this parameter, e.g.`/opt/IBM/tivoli/netcool/ncm/`. This will be the location where all `syslog` files will be saved to.

The OutOfBandChange daemon will always parse the `syslog` file specified in the `oobc.properties.xml` property file. If, as can happen in a Unix environment, the `syslog` file is rolled over by an administrative utility, then the OutOfBandChange daemon will also attempt to rollover to the new `syslog` file.

This rollover is predicated on the fact that typical `syslog` administrative scripts will follow a standard procedure:

1. Remove the oldest `syslog` file.
2. From oldest to newest, rename the files to a file name with a larger sequence number.
3. Open the new `syslog` file.

If, however, the OutOfBandChange daemon has the current `syslog` file open for reading while the `syslog` rollover occurs, there is no indication that this has occurred and therefore the OutOfBandChange daemon will be reading the now old `syslog` file.

Because of this scenario, the OutOfBandChange daemon will close its `syslog` file after a certain amount of time has expired with no new updates written to the log file. Then, it will open the `syslog` file again and wait again for the specified amount of time for updates written to the log file. If no updates occur within that time frame the file is closed and reopened again. This will repeat forever as long as there are no updates to the `syslog` file within the specified time frame and as long as the OutOfBandChange daemon is running.

If, however, a `syslog` file was rolled over from say`local6.log` to `local6.1.log` and a new `local6.log` file was created, the daemon will still have the old file open for read operations. Since nothing new is being written to the `local6.1.log` file, it will eventually be closed by the daemon. The daemon will then attempt to open up `local6.log` again, this time picking up the newly created (rolled) log file, and continue parsing.

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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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

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

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

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

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

IBM Corporation
958/NH04
IBM Centre, St Leonards
601 Pacific Hwy
St Leonards, NSW, 2069
Australia

IBM Corporation
896471/H128B
76 Upper Ground
London SE1 9PZ
United Kingdom

IBM Corporation
JBF1/SOM1
294 Route 100
Somers, NY, 10589-0100
United States of America

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 document 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.

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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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

Trademarks

IBM, the IBM logo, ibm.com®, Netcool®, Passport Advantage®, Tivoli®, the Tivoli logo and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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.

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

Linux® is a registered trademark of Linus Torvalds 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.

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

Index

Part Number:

Printed in the Republic of Ireland

(1P) P/N:

2023-4219-01