

NUU: Network Update Utility

User's Manual

Version 1.2

2006

Copyright

Copyright 2006 by Concurrent Computer Corporation. This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License Version 2 (GPLv2), which should be included with this software.

This program is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**. See the GNU General Public License for more details.

For other license arrangements, contact Concurrent Computer Corporation.

Disclaimer

The information contained in this document is subject to change without notice. Concurrent Computer Corporation has taken efforts to remove errors from this document, however, Concurrent Computer Corporation's only liability regarding errors that may still exist is to correct said errors upon their being made known to Concurrent Computer Corporation.

Trademark Acknowledgments

RPM is a trademark of Red Hat, Inc.

Contents

Command Line Options.	1-1
Graphical User Interface	1-1
Starting.	1-2
Main Window	1-3
Package List Controls.	1-3
Package List Content	1-5
Installing, Updating, and Removing.	1-6
Suite and Group Removal.	1-7
Install-only Packages.	1-7
Other Main Window Functions	1-8
Log Window	1-9
Package Description	1-10
Repository Configuration Editor	1-13
Repository List	1-14
Repository Configuration.	1-15

NUU (Network Update Utility) is a graphical user interface which allows a user to make modifications to the installed rpm™ packages on a system.

Its capabilities are:

- Install new packages or update existing packages by downloading them from a yum package repository
- Remove existing packages
- Configure sets of repositories from which to obtain new and updated packages
- Examine installed packages or packages available for installation or update
- Install, update, or remove groups of packages
- Install or update suites of packages

Command Line Options

Use the following command to start NUU:

nuu [*options*]

Usually, no options are necessary. But the following are available:

-r, --repo-editor

Start by bringing up the repository configuration editor. Use this to edit the repositories before attempting to read information from them.

-l, --log

Start with the log window displayed.

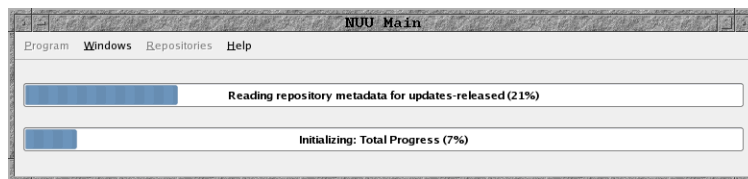
Graphical User Interface

Most graphical items provide *tool tips*. If unsure of the meaning of a particular user control or other item, simply move the mouse pointer over the item and let it hover there briefly. A tool tip will appear in a small transient window explaining the item.

Most dialogs support pressing **Enter** to perform the default action for that dialog. Similarly, they support pressing **Escape** to cancel an action (if appropriate) and close the dialog.

Starting

When NUU is started, it first initializes itself by downloading information from remote repositories (although this may be skipped if information was cached locally by a previous invocation of NUU or yum and that information is reasonably current) and then loading that information. If the number of packages installed or the number of packages available in configured repositories is very large, this may take a while. The initial window is abbreviated and displays two progress bars:



Frequently a NUU activity will be comprised of numerous sub-activities. The upper progress bar describes the sub-activity currently in progress and estimates the percentage of its progress completed. The lower progress bar describes the entire activity and estimates the percentage of its progress completed.

The progress bars will continue to exist after start-up to display other long-term activities.

Caveats

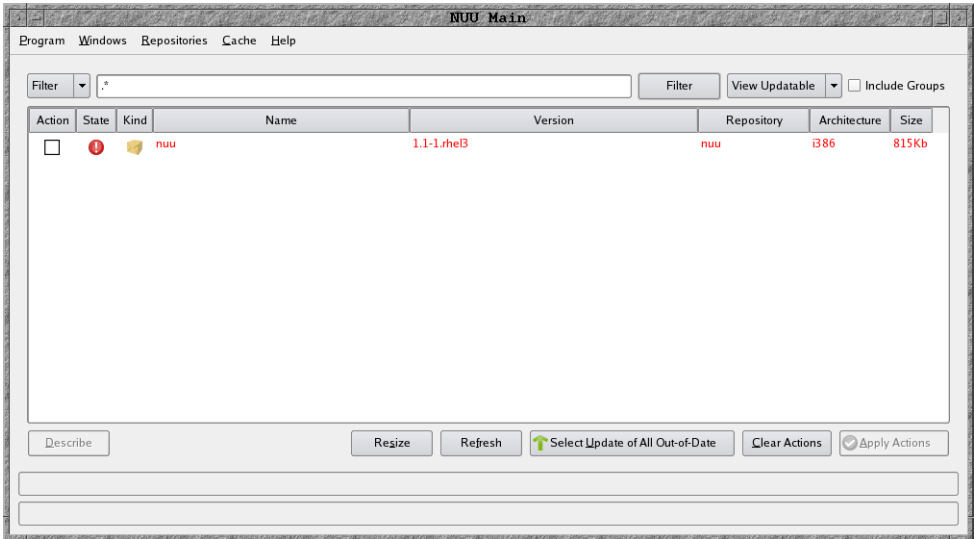
Not all sub-activities can show incremental progress. Sub-activities which do not will display only 0% and 100% progress. They still are displayed in the upper progress bar to give an indication of the sub-activity in progress, even though the percentage of that progress cannot be estimated.

It is not always possible to know the relative proportions of the durations of heterogeneous sub-activities. The lower progress bar uses a set of heuristics to estimate the relative proportions of many heterogeneous sub-activities.

Once NUU is initialized, additional items will be added to the initial window and it will be a fully-functioning Main Window.

Main Window

NUU’s main window displays packages which are installed, or are available for installation or update from the configured repositories. It allows the user a number of ways to determine which packages are displayed, and it allows the user to control which packages will be installed, updated, or removed. It also allows the user to display information about each package. Following is an example of the window’s initial appearance:

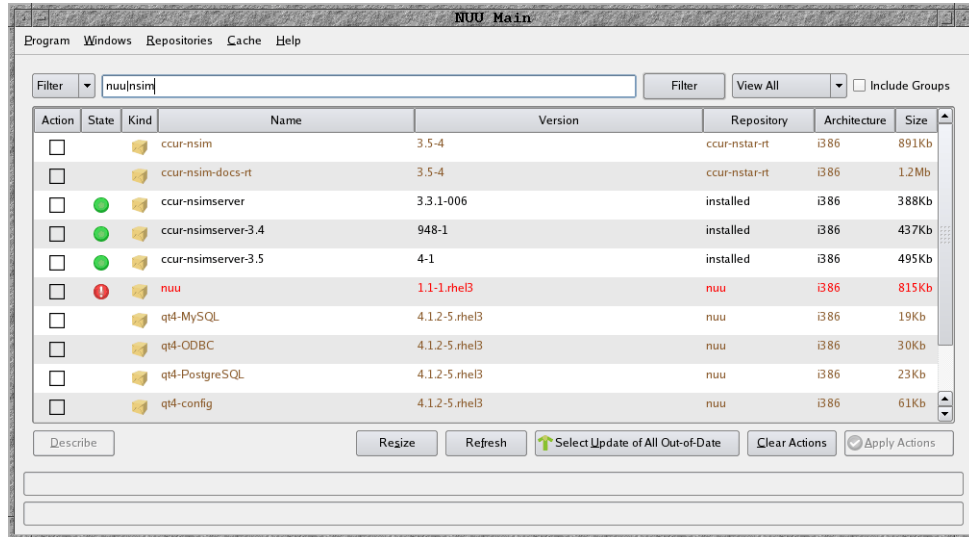


Package List Controls

Initially, NUU displays only updatable packages. That is, it displays packages which are installed but for which newer versions are available. This is evidenced by the control near the upper right which reads **View Updatable**. Updatable packages are easily identifiable because their State column contains a red disc with an exclamation point and because their text also is in red.

Other possible values for this control are **View Installable**, **View Removable**, and **View All**. Installable packages are those for which no version is installed but one is available to be installed. They are identified by the absence of a graphic in their State column, and because their text is tan. Removable packages are any packages which currently are installed. They include all updatable packages, as described above. They also include packages for which a version is installed and no newer version is available in any configured repository. Packages which are installed and for which no newer version is available

are identified by a green disc in their State column and because their text is black. The **View All** setting displays packages in all states. Following is an example:



The **View All** control frequently displays hundreds or thousands of packages. The text field above the package list is useful for selecting a smaller set of packages to display. It may be used in either of two ways: **Filter** or **Search**. In the above example, the **Filter** mode is used. The control in the upper left is set to **Filter**, and a regular expression is specified in the text field. In the above case, the regular expression `nuu|nsim` was used. This regular expression is applied to all columns visible in the package list as if they were separated by whitespace. So it is possible to filter based not only on the Name, but also on Version, Repository, Architecture, Size, or combinations thereof. To apply the regular expression, either press the **Filter** button to the right of the text field, or press **Enter** in the text field itself.

The text field's other mode is **Search**. It is selected by changing the control in the upper left to **Search**. In this mode, the regular expression is applied not just to the fields displayed in the package list, but to every field in the rpm package. This includes things like the Summary, Description, Filenames, the rpm Provides, Requires, Conflicts fields, and several others. A list of the fields appears in Table 1-5, "Package Description Fields," on page 1-11 and in Table 1-6, "Group Description Fields," on page 1-12. This mode of controlling the packages displayed is significantly slower than the **Filter** mode. Some of these fields are not even loaded by default. So, using the **Search** mode for the first time may require NUU to load the additional information. Nonetheless, it is very useful if used judiciously.

NOTE

The **Search** mode is significantly slower than the **Filter** mode.

The last remaining control is in the upper right and is a check box labeled **Include Groups**. Groups are discussed in Table 1-1, "Package List Columns," on page 1-5 under **Kind**, and in "Suite and Group Removal" on page 1-7. If the box is checked, then groups are displayed in the package list; otherwise, they are not.

Package List Content

Each line in the package list describes a package, a suite of packages or a group of packages. However, throughout this document, frequently the term *package* is used for brevity. In the case of the package list, it implies not just packages, but also suites and groups. Each column describes an aspect of that package. The columns are as follows:

Table 1-1. Package List Columns

Column Name	Meaning
Action	This controls whether the package will be installed, updated, or removed. It is discussed in “Installing, Updating, and Removing” on page 1-6.
State	<p>The absence of a graphic indicates that the package is not installed, but is available for installation.</p> <p>A green disc indicates that the package is installed and no newer version exists in any of the configured repositories.</p> <p>A red disc with an exclamation point indicates that the package is installed but that a newer version exists in one of the configured repositories.</p>
Kind	<p>A graphic of a tan single package indicates that this an ordinary rpm package.</p> <p>A graphic of 3 tan packages indicates that this is a suite. A suite is a meta-package which has no real content but has dependencies on a number of other packages. It is used to install or update a number of related packages quickly and easily.</p> <p>A graphic of 3 grey packages indicate that this is a group. A group is similar to a suite but is more limited in that it cannot specify requirements on particular versions. However, removal of whole groups is supported, whereas it is not for suites.</p>
Name	This is the name of the package.
Version	<p>For an updatable package, this is the version number of the newer package which is available for installation. The version number of the installed package can be seen by moving the mouse over the field and allowing it to hover there until a tool tip appears. The tool tip will describe both version numbers.</p> <p>For an installable package, this is the version number of the package which is available for installation.</p> <p>For an installed package for which no newer version exists, this is the version number of the installed package.</p>





Table 1-1. Package List Columns

Column Name	Meaning
Repository	For an installed package with no newer version, this is just the string <code>installed</code> . For other packages, it is the identifier of the repository from which the package or newer version is available.
Architecture	This is the architecture of the package. For instance, <code>i386</code> or <code>x86_64</code> .
Size	This is the size of the package.

Installing, Updating, and Removing

The **Action** column is used to specify whether a package should be installed, updated, or removed, or that no action should be taken with respect to the package. By default, an empty box is displayed, and this indicates that no action should be taken. The possible actions are determined by the package's state, and are as follows:

Table 1-2. Package Actions

Graphic	Action	Availability
	None	Available for all packages.
	Install	Available for packages which are not installed currently.
	Update	Available for packages which are installed but for which a newer version is available.
	Remove	Available for packages which are installed (regardless of whether or not any newer version is available).

To change the action for a package, simply press mouse button 1 (usually the left mouse button) in the **Action** column for the particular package. It will cycle through all actions which are applicable given the package's state. Alternatively, press mouse button 3 (usually the right mouse button) in the same location, and pop-up menu will appear allowing selection of the desired action directly.

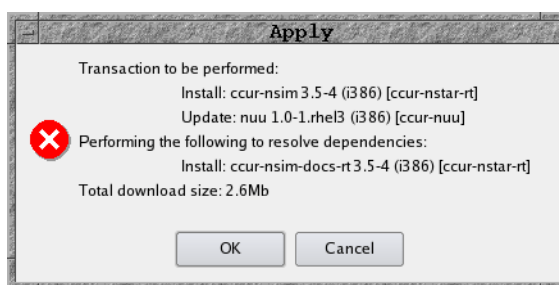
Selecting an action will not have an immediate effect beyond changing the graphic in the **Action** column. The actions will not be performed until they are applied. This is so that actions for multiple packages can be chosen and then applied all at once in a single transaction.

As a convenience for one of the most common situations, a button near the bottom labeled **Select Update of All Out-of-Date** is provided. It sets the action to **Update** for all updatable packages.

Another button, labeled **Clear Actions**, is provided to reset all actions back to None.

The **Apply Actions** button is used to perform the actions, once all desired actions have been selected.

When the **Apply** button has been pressed, if any installations or updates were selected, NUU will download rpm headers to determine details about the rpm packages to be installed or updated. It then will attempt to resolve any dependencies. Assuming this is successful, a dialog like the following will appear describing the transaction to be performed:



In this example, the **ccur-nsim** package was selected for installation and the **nuu** package was selected for update. NUU determined that this would cause an unresolved dependency because **ccur-nsim** requires the **ccur-nsim-docs** capability. It resolved that by suggesting that the **ccur-nsim-docs-rt** package also should be installed, as it provides that capability.

The user can press **OK** to accept the transaction presented by NUU or can press **Cancel** to return to the **Main Window** to do something different.

If **OK** is selected, NUU will download any required rpm packages, and then will install or update them; and if any removals were selected, it will remove those packages.

If any fatal errors are detected either during the attempt to resolve dependencies, or during the actual transaction, an error dialog will be displayed, and the apply will be aborted.

Suite and Group Removal

One of the main differences between suites and groups concerns their removal. If a suite is removed, it does not remove the packages on which it depends. This derives from the fact that a suite really is just a special rpm package.

However, if a group is removed, all its packages are removed.

Install-only Packages

Some packages are treated as *install-only*. NUU will only install them and never update them. This is regardless of the fact that the Update graphic is available for them in the package list's Action column, and that the Apply dialog can claim that they will be updated.

By default the set of install-only packages includes those with the following names:

- kernel
- kernel-smp
- kernel-enterprise
- kernel-debug
- kernel-unsupported

The *install-only* behavior can create duplicate packages on the system. The reason for this is that, as part of a kernel update, the currently running kernel might be removed. This can be problematic for the running system. Furthermore, the kernel is the most critical component of the system, and a bug in the newer version of the kernel could render the system unable to boot. Having the older kernel available is useful in an emergency, since it presumably worked and can be booted so that problems can be fixed.

Other Main Window Functions

The Main Window has additional menu items.

Under the **Program** menu, the menu item **Quit** will cause NUU to exit. Alternatively, **Ctrl+Q** may be pressed.

Under the **Cache** menu, there are several menu items:

Table 1-3. Cache Menu Items

Menu Item	Meaning
Clean Headers	Clean RPM package headers containing basic information about packages that are or once were available.
Clean Packages	Clean RPM packages for packages that are or once were available. This is useful for recovering disk space consumed by RPM packages that already have been installed.
Clean XML Metadata	Clean XML metadata containing information about packages provided by each repository. This, along with Clean Pickled Cache , is useful forcing a download of new available package lists, in case a new package was made available very recently.
Clean Pickled Data	Clean pickled data containing information about packages provided by each repository. This, along with Clean XML Metadata , is useful forcing a download of new available package lists, in case a new package was made available very recently.
Clean All	Clean all cached data. It is equivalent to using all of the above items.

Under the **Help** menu, the **About** menu item will display information about NUU and its version number.

The Main window has additional buttons along the bottom.

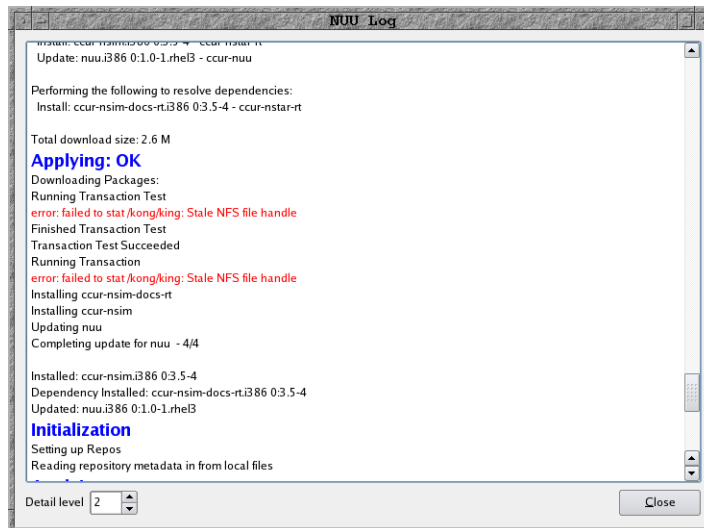
The **Refresh** button causes NUU to discard all information that it has loaded and to re-load that information.

The **Resize** button causes NUU to examine the fields for all packages that it has loaded and to resize the package list and the entire Main Window so that no fields are truncated. This often is convenient after a Refresh.

The **Describe** button is used to provide more detailed information about a selected package. Alternatively, pressing mouse button 3 (usually the right mouse button) while on the line for any package and in any column other than Action, will cause a pop-up menu to appear from which a **Describe** menu item can be selected. This is described in “Package Description” on page 1-10.

Log Window

Sometimes it may be desirable to see details of a transaction that is taking place or that has taken place. The Log Window is provided for this purpose. It is opened using by selecting the **Windows** menu and then selecting the **Log** menu item. Alternatively, it can be opened by pressing **Ctrl+L**. A window like the following will appear:



In this window, items are presented in various colors to ease interpretation. Table 1-4 describes them.

Table 1-4. Log Window Colors

Color	Meaning
Blue	High-level action such as Initialization or Applying
Black	Normal log entries
Green	Output from rpm scriptlets (stdout)
Red	NUU, rpm, or yum errors; Errors from rpm scriptlets (stderr)
Orange	Warnings
Tan	Debug information

In the example provided, two NFS errors were induced to demonstrate the colors. But those errors were not fatal, and so the transaction completed successfully. Nonetheless, because they were errors, they appeared in red.

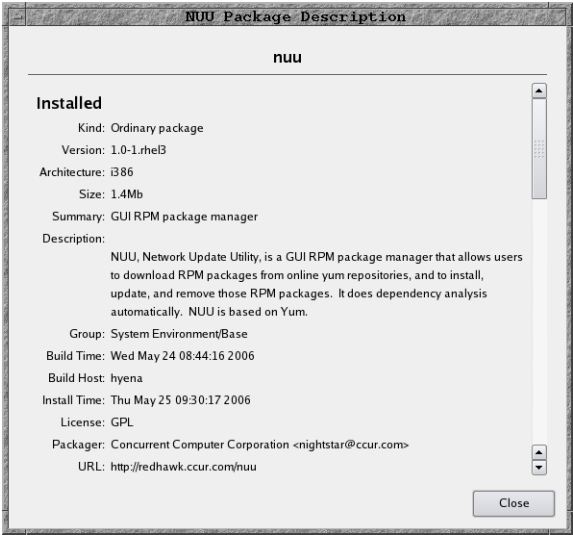
The Log Window has a control in the lower left called **Detail Level**. It corresponds to yum's debugging level. It defaults to the value 2, which provides a reasonable amount of information. It can be raised to see more detailed log information, or lowered to see less detailed log information. Higher levels of detail will show information which might be considered debug information.

The Close button can be used to close the Log Window.

Package Description

The Main Window's **Describe** button is used to provide more detailed information about a selected package. Alternatively, pressing mouse button 3 (usually the right mouse button) while on the line for any package and in any column other than Action, will cause

a pop-up menu to appear from which a **Describe** menu item can be selected. In either case, a **Package Description** dialog will appear similar to the following:



Some of the information displayed there is the same as in the package list described in Table 1-1, “Package List Columns,” on page 1-5. But additional information also is supplied, if available and applicable:

Table 1-5. Package Description Fields

Field	Meaning
Repository	The identifier for the repository from which a package is available. A Details button also is available which pops up a small window with additional information about the repository.
Summary	A one-line summary of the purpose of the package.
Description	A detailed description of the purpose of the package.
Group	A description of a type of functionality provided, used to group packages together. Nominally the set of groups is well-defined in the <code>/usr/share/doc/rpm-version/GROUPS</code> or <code>/usr/share/doc/packages/rpm-version/GROUPS</code> file. But packages do deviate from those sets. In particular the special value <code>Suite</code> is used by NUU to distinguish a suite from an ordinary package.
Vendor	The identity of the vendor which packaged the rpm package.
Build Time	The date and time at which the rpm package was created.
Build Host	The name of the host system on which the rpm package was created.
Install Time	The date and time at which the rpm package was installed on this system.
License	A short textual code indicating the licensing terms for the package. Common values are GPL, LGPL, BSD, and Commercial.

Table 1-5. Package Description Fields

Field	Meaning
Packager	The identity and e-mail address of the person or company that produced the rpm package. This frequently is <u>not</u> the same as the person or company who wrote the software.
URL	A URL (typically a web address) for the software.
Relocations	Any filename path prefixes in the rpm package which can be or were relocated.
Provides	A list of the capabilities provided by the rpm package. Note, however, that a package always provides its own name and version as one capability and that is not listed here.
Requires	A list of the capabilities required by the rpm package, along with any particular version requirements for those capabilities.
Conflicts	A list of the capabilities that conflict with this rpm package, possibly with information about only specific versions which conflict.
Obsoletes	A list of the capabilities that are obsoleted by this rpm package.

In addition, two special fields are provided: **Files** and **Change Log**. Each has only a **Show** button, possibly also saying (**must load extra data**). The contents of these fields are not displayed because they potentially can be very long. Also, for packages that are available but not installed, they require loading additional data from remote repositories that is not downloaded or loaded by default. If this information has not been loaded in the current NUU session, then the buttons will contain the text (**must load extra data**), and pressing the button will necessitate loading the extra data, including downloading it from repositories if it isn't cached locally from a previous NUU session.

The **Files** field contains a list of pathnames for every file in the rpm package.

The **Change Log** field contains a list of Change Log entries for the rpm package. Note, however, that not all rpm packages contain this information.

The rpm fields listed above are for ordinary packages and package suites. The descriptions for groups are significantly different, as they are not really rpm packages. Their additional fields are as follows:

Table 1-6. Group Description Fields

Field	Meaning
Required Groups	A list of other groups required by this group.
Default MetaPkgs	A list of other groups installed by this group by default.
Optional MetaPkgs	A list of other groups which can be installed optionally as part of this group.
Mandatory Packages	A list of packages required by this group.

Table 1-6. Group Description Fields

Field	Meaning
Default Packages	A list of packages installed by this group by default.
Optional Packages	A list of packages which can be installed optionally as part of this group.
Conditional Packages	A list of packages which can be installed conditionally as part of this group.

If a **Package Description** dialog is created for a package that is installed but updatable, then it displays information for the updatable version in the upper portion of the dialog, and displays information for the currently installed version in the lower portion. The information for the installed version is abbreviated to exclude fields which are unlikely to differ between the two versions.

Repository Configuration Editor

To use NUU to edit a system's repository configuration, select the **Repositories** menu and then **Edit Configuration** menu item. Alternatively, it can be opened by pressing

Ctrl+E. This will cause the NUU Repository Configuration Editor to open, with an appearance like the following:

Identifier	Name
<input checked="" type="checkbox"/> ccur-endigen	Concurrent Endigen
<input checked="" type="checkbox"/> ccur-fortran	Concurrent Fortran 77
<input checked="" type="checkbox"/> ccur-maxada	Concurrent MAXAda
<input checked="" type="checkbox"/> ccur-maxaxi	Concurrent AXI for MAXAda
<input type="checkbox"/> ccur-nstar-lx	Concurrent NightStar LX
<input checked="" type="checkbox"/> ccur-nstar-rt	Concurrent NightStar RT
<input checked="" type="checkbox"/> ccur-nstar-rt-devel	Concurrent NightStar RT Development
<input checked="" type="checkbox"/> ccur-nstar-rt-internal	Concurrent NightStar RT Internal
<input checked="" type="checkbox"/> ccur-nuu	NUU Updates

Repository Configuration:

Identifier:

Name:

URLs:

Mirror list URL:

Buffer Authentication:

Name:

Password:

Repository Options:

Enabled:

Check GPG Keys:

Other Options:

Configuration File (expert):

File: Sort Position:

Repository List

The top portion of this editor displays a list of all repositories configured on the system. There are three columns displayed. The first is a checkbox which indicates whether or not the repository is enabled during this session. The second is the identifier for the repository. And the last is a descriptive name for the repository.

To enable or disable a repository during this session, simply press mouse button 1 (typically the left mouse button) in the first column for the particular repository. This toggles the checkbox.

To edit the configuration for an existing repository, simply click on any other column for the particular repository. This will enable the bottom portion of the editor where configuration parameters are shown and can be modified for the particular repository.

To add a new repository configuration, either of the Add or Copy buttons can be pressed. If Add is pressed, it adds a blank new repository configuration to the list, selects it, and enables the bottom portion of the editor where the new repository configuration can be edited. If the Copy button is pressed, it copies the currently selected repository configura-

tion into a new repository configuration in the list, selects it, and enables the bottom portion of the editor where the new repository configuration can be edited.

To remove existing repository configurations, select them, and press the **Remove** button.

Repository Configuration

After selecting a repository configuration to be edited, or pressing **Add** or **Copy**, the bottom portion of the **Repository Configuration Editor** becomes enabled and displays the current parameters of the selected repository configuration. For example, the window might appear as follows:

The screenshot shows the **NUU Repository Configuration Editor** window. It features a table with two columns: **Identifier** and **Name**. The table lists several repository configurations, with checkboxes in the **Identifier** column. The **ccur-nuu** configuration is selected. Below the table are buttons for **Add**, **Copy**, and **Remove**.

Below the buttons, the **Repository Configuration:** section displays the details for the selected **ccur-nuu** configuration:

- Identifier:** ccur-nuu
- Name:** NUU Updates
- URLs:** http://redhawk.ccur.com/nuu/rhel3/i386/
- Mirror list URL:** (empty field)
- Buffer Authentication:**
 - Name:** (empty field)
 - Password:** (empty field)
- Repository Options:**
 - Enabled:** Default (Yes) (dropdown menu)
 - Check GPG Keys:** Default (No) (dropdown menu)
 - Other Options:** (empty text area)
- Configuration File (expert):**
 - File:** /etc/yum.repos.d/ccur-nuu.repo
 - Sort Position:** 1.0 (dropdown menu)

At the bottom of the window are buttons for **Reset**, **Reset All**, **Cancel**, **Apply**, and **OK**.

Those parameters can be modified. The parameters are as follows:

Table 1-7. Repository Configuration Parameters

Parameter	Meaning
Identifier	An identifier for the repository, used when mentioning the repository elsewhere in NUU.
Name	A descriptive name for the repository.
URLs	A list of URLs, one per line. All URLs in the list should refer to repositories which are mirrors of one another. A responsive URL will be chosen from the set arbitrarily.
Mirror list URL	A URL to a file which contains a list of URLs which all are mirrors of one another.
Buffer Authentication:	For repositories which use buffet authentication:
Name or Login or E-mail	A name, login, e-mail address or any other identifier appropriate for the repository to identify a valid user.
Password or Order Number	An identifier to authenticate that the user really is valid.
Repository Options:	
Enabled	Whether or not this repository is enabled. Unlike the checkbox in the Repository List, this parameter is persistent from session to session. If this parameter is changed, the value of the checkbox in the Repository List will change to reflect it after all changes to this repository configuration are accepted.
Check GPG Keys	Whether or not GPG Keys in rpm packages are checked against keys installed in the rpm database for correctness. This option can be used only if GPG Keys have been installed in this system's rpm database manually, and if all rpm packages from the repository contain GPG Keys.
Other Options	Any other valid option for a repository configuration. See yum.conf (5) for a complete list of options. This field is provided for experts.

Table 1-7. Repository Configuration Parameters

Parameter	Meaning
Configuration File:	
File	The name of the configuration file in which this repository configuration exists. It must be either <code>/etc/yum.conf</code> or a file of the form <code>/etc/yum.repos.d/name.repo</code> . If left blank, the identifier for the repository will be used to create a name. This field is provided for experts.
Sort Position	This number specifies a sort position for the repository configuration within its configuration file. It is meaningful if multiple repository configurations exist within a single file, and controls the order in which they appear. It is provided for experts.

NUU can detect several mistakes in repository configurations. Indications for these will appear in messages in the bottom left of the **Repository Configuration Editor**. In particular, each repository configuration must have an identifier. It also must have either at least one URL or a Mirror List URL.

Changes made in the Repository Configuration are not reflected immediately in the Repository List. Similarly, the checkbox in the Repository List does not change immediately based on a change in the Enabled parameter. Changes appear in the Repository List only after they are accepted, which is done by selecting a different repository configuration in the Repository List, or by pressing **Apply** or **OK**.

The **Reset** button undoes changes to the currently selected repository configuration, reverting its parameters back to the values they had when it last was selected or when **Apply** was last pressed, whichever was more recent. In short, it undoes any changes which have not yet been accepted.

The **Reset All** button, on the other hand, undoes changes to all repository configurations, reverting them all the way back to their values since the last **Apply** or since the **Repository Configuration Editor** was opened, whichever was more recent.

The **Cancel** button operates like the **Reset All** button, but also closes the **Repository Configuration Editor**.

The **Apply** button actually changes the repository configurations on the system. Changes have no effect until and unless **Apply** (or **OK**) is pressed.

The **OK** button operates like the **Apply** button, but also closes the **Repository Configuration Editor**.

If any changes were made to the system's repository configuration, when the **Repository Configuration Editor** is closed, a Refresh of the package list will occur automatically.

The configuration files also can be edited manually. They are the same files as used by **yum(8)**. They are `/etc/yum.conf` and files of the form `/etc/yum.repos.d/name.repo`. See **yum.conf(5)** for details.

