NightSim RT User's Guide

Version 4.1

(RedHawk™ Linux®)

concurrent

0898480-070
July 2008

The information contained in this document is believed to be correct at the time of publication. It is subject to change without notice. Concurrent Computer Corporation makes no warranties, expressed or implied, concerning the information contained in this document.

To report an error or comment on a specific portion of the manual, photocopy the page in question and mark the correction or comment on the copy. Mail the copy (and any additional comments) to Concurrent Computer Corporation, 2881 Gateway Drive, Pompano Beach, FL 33069-4324. Mark the envelope **"Attention: Publications Department**." This publication may not be reproduced for any other reason in any form without written permission of the publisher.

Preface

Scope of Manual

This guide is designed to assist you in getting started with the use of NightSim™. NightSim is a real-time NightStar™ RT tool that provides a graphical user interface to the frequency-based scheduler and performance monitor services.

Structure of Manual

This manual consists of five chapters, four appendixes, a glossary, and an index. A brief description of the chapters and appendixes is presented as follows.

- Chapter 1 introduces you to the concepts and components of NightSim, a real-time tool that is part of the NightStar™ development environment.
- Chapter 2 provides an overview of the primary factors that need to be taken into account prior to running NightSim.
- Chapter 3 explains the procedures for beginning and ending a NightSim session and explains how to use the Menu bar and toolbar.
- Chapter 4 describes the frequency-based scheduler and introduces the components of NightSim's Scheduler and Processes areas.
- Chapter 5 describes the performance monitor and introduces the components of NightSim's Monitor area.
- Appendix A discusses the NightStar License Manager (NSLM) and how to obtain and install licenses. It also discusses approaches for dealing with a firewall either on the system acting as the license server or on a system hosting the NightStar RT tools.
- Appendix B describes dependencies of certain features on the RedHawk kernel.
- Appendix C discusses the format of configuration files used by NightSim.

The Glossary contains definitions of technical terms that are important to understanding the concepts presented in this book.

The Index contains an alphabetical reference to key terms and concepts and numbers of pages where they occur in the text.

Syntax Notation

The following notation is used throughout this manual:

italic

Books, reference cards, and items that the user must specify appear in *italic* type. Special terms may also appear in *itali*cs.

`list bold`

User input appears in **`list bold`** type and must be entered exactly as shown. Names of directories, files, commands, options and system manual page references also appear in **`list bold`** type.

`list`

Operating system and program output such as prompts and messages and listings of files and programs appear in `list` type.

`[ ]`

Brackets enclose command options and arguments that are optional. You do not type the brackets if you choose to specify such options or arguments

Referenced Publications

The following publications are referenced in this document:

0890514	*NightBench User's Guide*
0898007	*Real-Time Clock and Interrupt Module (RCIM) User's Guide*
0898004	*RedHawk Linux™ User's Guide*
0898008	*NightStar RT Installation Guide*
0898009	*NightStar RT Tutorial*
0898395	*NightView RT User's Guide*
0898398	*NightTrace RT User's Guide*
0898465	*NightProbe RT User's Guide*
0898515	*NightTune RT User's Guide*
0898537	*MAXAda for Linux Reference Manual*

Contents

Tables

1
Overview

This chapter introduces you to the concepts and components of NightSim, a real-time tool that is part of the NightStar development environment. NightSim provides a graphical user interface to the major functions associated with the *frequency-based scheduler* and the *performance monitor.*

Overview of NightSim

NightSim allows you to perform the entire range of functions associated with frequency-based scheduling. You can perform the major functions of configuring a scheduler, setting up a interrupt source, scheduling programs, saving and restoring a scheduler configuration, running a simulation, and viewing scheduling data. The principal features of the *frequency-based scheduler* are described in the section that follows.

NightSim also allows you to perform the entire range of functions associated with the *performance monitor*. You can perform the major functions of selecting a scheduler, clearing existing performance monitor values, enabling and disabling performance monitoring, including and excluding interrupt time, and viewing performance monitor values. The principal features of the performance monitor are described in “Overview of the Performance Monitor” on page 1-2.

In addition, NightSim supports the use of interrupt sources, including Distributed Interrupts devices on the RCIM (Real-Time Clock and Interface Module) board. An addition capability is NightSim's ability to configure a frequency-based scheduler on any system in your network, not just the *NightSim host*.

NightSim is also integrated with other NightStar tools. NightSim may be invoked directly from the Tools menu of any NightStar RT tool, and NightSim provides access to every other NightStar RT tool in the same way. NightSim also provides the ability to schedule your programs under the control of the NightView™ Source-Level Debugger, making it easy to gain control of scheduled programs in order to set breakpoints or monitorpoints early on in the execution of the program, before it attaches to the frequency based scheduler.

Overview of the Frequency-Based Scheduler

The *frequency-based scheduler*, or FBS, is a task synchronization mechanism that allows you to run processes at specified frequencies. Frequencies can be based on high-resolution clocks, an external interrupt source, or completion of a cycle. The frequency-based

scheduler provides a mechanism for initiating processes at the specified frequency. The processes are then scheduled via the standard priority-based scheduler.

When used in conjunction with the performance monitor, the FBS can be used to determine the best way of allocating processors to various tasks for a particular application. "Using the Scheduler Area" on page 4-1 fully describes the features and capabilities of the frequency-based scheduler.

Overview of the Performance Monitor

The performance monitor is a mechanism that allows you to monitor use of the CPU by processes that are scheduled on a frequency-based scheduler. The values obtained can help in determining whether processes need to be redistributed among processors for improved load balancing and processing efficiency. You can obtain performance monitor values by process or processor and can specify whether time spent servicing interrupts should be included or excluded from the values. The performance monitor also makes it possible to monitor a processor's *idle time*. By examining the amount of idle time on each processor, you can identify the processors that have the lightest load and determine the additional amount of CPU time that can be used for scheduling real-time processes. "Using the Monitor Area" on page 5-1 fully describes the features and capabilities of the performance monitor.

Overview of the Graphical User Interface

NightSim's graphical user interface consists of a single window with areas that provide access to the key operations associated with frequency-based scheduling and performance monitoring. An overview of the menu selections and areas is provided in the sections that follow.

When NightSim is launched without arguments, the following window displays.

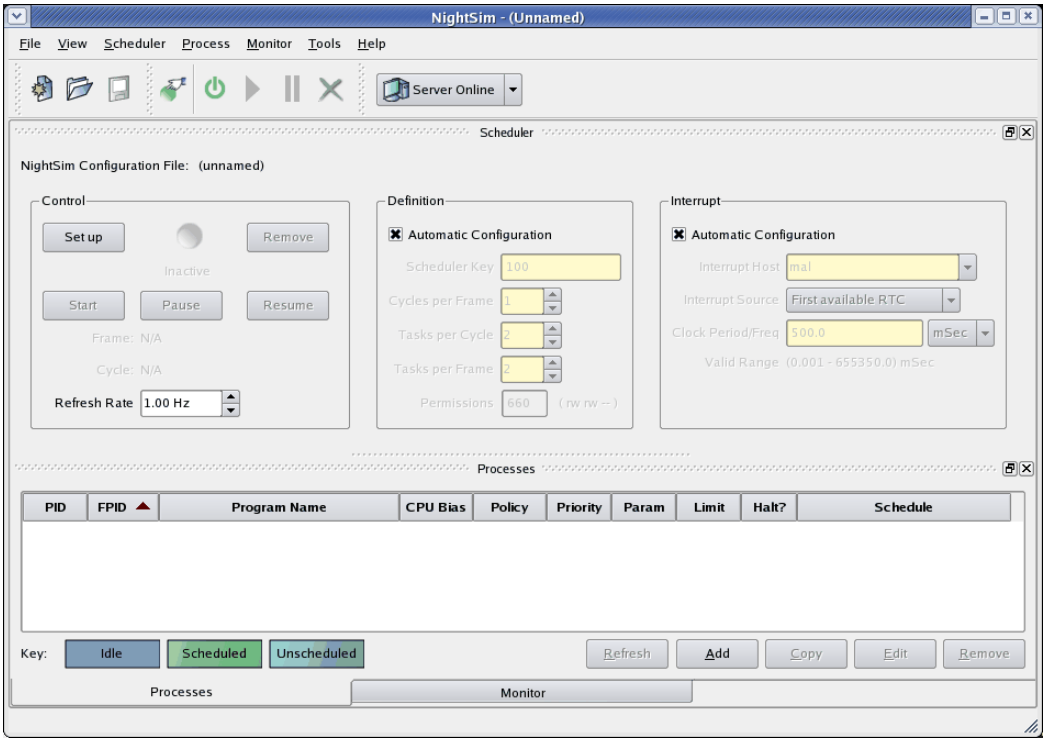


Figure 1-1. The NightSim Window

The following areas are available for viewing within this window.

The Scheduler area consists of three sections:

- Control

  This area allows you to create and remove a *frequency-based scheduler* and start, stop and resume a simulation. The status of the simulation run is also provided. Refer to the section “Using the Scheduler Control Area” on page 4-12 for a full description.

- Definition

  This area allows you to configure a *frequency-based scheduler*, specify maximum values for the scheduler, and specify the permissions that are to be associated with it. Refer to the section “Using the Scheduler Definition Area” on page 4-16 for a full description.

- Interrupt

  This area allows you to select the *interrupt host* and the interrupt source that is to be used for the scheduler. Refer to the section “Using the Scheduler Interrupt Area” on page 4-17 for a full description.

The Processes area allows you to select processes, add, edit, copy and remove processes from a scheduler and view data for the scheduled processes. Refer to the section "Using the Processes Area" on page 4-20 for a full description.

The Monitor area appears by clicking on the Monitor tab at the bottom of the screen or by selecting Monitor from the View menu. Performance Monitor data for the scheduled processes can be viewed in this area. Refer to "Using the Monitor Area" on page 5-1 for a full description.

NOTE

It is important to note that each of the three areas of the NightSim window can be "torn off" to reside in its own window separate from NightSim by clicking on the dashed line at the top of the area and dragging to another location on the desktop.

The toolbar at the top of the NightSim window contains icons for commonly used actions. The toolbar is described in the section "The Toolbar" on page 3-11.

Overview of RCIM Support

NightSim supports the Real-Time Clock and Interrupt Module (RCIM). An RCIM is a hardware module attached to a *target system* which contains the following interrupt-generating devices:

- real-time clocks (RTCs)
- edge triggered interrupts (ETIs)
- programmable interrupt generators (PIGs)

Any of these devices may be used as the interrupt source for an FBS.

RCIMs may be connected via RCIM cables such that interrupts from one RCIM may be distributed to other systems in the RCIM chain.

Each interrupt device on an RCIM may be configured as an *RCIM-distributed* or normal device. By default, each interrupt device is available for use on the attached target system. If the interrupt device is configured as an RCIM-distributed device, interrupts are also distributed between target systems using the RCIM cable.

NightSim provides the ability to create, control and monitor performance of individual schedulers distributed across multiple SBC's and/or target computers in an RCIM chain utilizing an RCIM-distributed interrupt source.

For more information see the *Real-Time Clock and Interrupt Module (RCIM) User's Guide (0898007).*

2
Establishing the NightSim Environment

This chapter provides an overview of the primary factors that need to be taken into account prior to running NightSim. These include *system configuration requirements* and *user requirements.*

Configuration Requirements

NightSim requires the Frequency Based Scheduler to be installed on the target system. The FBS is an optional portion of RedHawk Linux.

Typically, the default installation of the FBS is sufficient for NightSim activities. Refer to the *RedHawk Frequency-Based Scheduler (FBS) User's Guide (0898005)* for more information on FBS configuration.

NightSim schedulers that are driven by an interrupt source require an RCIM device; a real-time clock, an edge-triggered interrupt, or a distributed interrupt source. If you do not have an RCIM device, you can still use NightSim's end-of-process scheduling capability to drive your simulation. See End of Cycle scheduling for more information.

User Requirements

To use NightSim, you must also ensure that certain user requirements related to remote shell access, environment variables, and capabilities are met. These requirements are described in the sections that follow.

Remote Shell Access

`rsh(1)` access is only required if you configure NightSim to run your scheduled process under an **`xterm(1)`**.

NightSim will run the **`xterm`** command on the NightSim host if the target is accessible from the host via **`rsh`**. If the **`xterm`** command is available on the target system, it is not necessary to enable **`rsh`** access, because NightSim will execute the **`xterm`** command on the target system.

To enable the use of **`rsh`**, ensure that a login exists on the remote target with the same username as on the NightSim host. In addition, the **`.rhosts`** file in that user's home

directory on the target system must contain an entry for that username and the NightSim host.

Your target system must also be properly configured to accept **rsh** command execution. See your target system documentation for details on how to enable **rsh** access to your target system.

See the **rsh(1)** man page for more details.

Capabilities

Some of the operations associated with the *frequency-based scheduler* and the *performance monitor* require certain capabilities such as CAP_SYS_NICE.

Linux provides a means to grant otherwise unprivileged users the authority to perform certain privileged operations. The Pluggable Authentication Module (see **pam_capability(8)**) is used to manage sets of capabilities, called *roles*, required for various activities.

Linux systems should be configured with an fbscheduser role which provides the CAP_SYS_NICE and CAP_DAC_OVERRIDE capabilities.

Edit **/etc/security/capability.conf** and define the fbscheduser role (if it is not already defined) in the "ROLES" section:

```
role fbscheduser CAP_SYS_NICE CAP_DAC_OVERRIDE
```

Additionally, for each NightSim user on the target system, add the following line at the end of the file:

```
user   username     fbscheduser
```

where *username* is the login name of the user.

If the user requires capabilities not defined in the fbscheduser role, add a new role which contains fbscheduser and the additional capabilities needed, and substitute the new role name for fbscheduser in the text above.

In addition to registering your login name in **/etc/security/capability.conf**, certain files under the **/etc/pam.d** directory must also be configured to allow capabilities to be activated.

To activate capabilities, add the following line to the end of selected files in **/etc/pam.d** if it is not already present:

```
session required pam_capability.so
```

The list of files to modify is dependent on the list of methods that will be used to access the system. Table 2-1 presents a recommended configuration that will grant capabilities to users of the services most commonly employed in accessing a system.

Table 2-1. Recommended /etc/pam.d Configuration

/etc/pam.d File	Affected Services	Comment
remote	telnet rlogin rsh (when used w/o a command)	Depending on your system, the **remote** file may not exist. Do not create the **remote** file, but edit it if it is present.
login	local login (e.g. console) telnet* rlogin* rsh* (when used w/o a command)	*On some versions of Linux, the presence of the **remote** file limits the scope of the **login** file to local logins. In such cases, the other services listed here with **login** are then affected solely by the **remote** configuration file.
rsh	rsh (when used with a command)	e.g. **rsh system_name a.out**
sshd	ssh	You must also edit **/etc/ssh/sshd_config** and ensure that the following line is present: **UsePrivilegeSeparation no**
gdm	gnome sessions	
kde	kde sessions	

If you modify **/etc/pam.d/sshd** or **/etc/ssh/sshd_config**, you must restart the **sshd** service for the changes to take effect:

```
service sshd restart
```

In order for the above changes to take effect, the user must log off and log back onto the target system.

NOTE

To verify that you have been granted capabilities, issue the following command:

```
/usr/sbin/getpcaps $$
```

The output from that command will list the roles currently assigned to you.

3
Getting Started

This chapter explains the procedures for beginning and ending a NightSim session. It also explains how to get help and how to use accelerators.

Invoking NightSim

The NightSim tool is available on your system as **/usr/bin/nsim**.

The syntax for executing **nsim** is as follows:

nsim [**-h**|**--help**] [**-v**|**--version**] [**-offline**]
[**-s** [*targetname*,]*key* |**--scheduler**=[*targetname*,]*key*]
[**-t** *targetname* |**--target**=*targetname*]
[**-***Xoption* ...] [[**-f**] *file*]

Options are described as follows:

-h |**--help**

Allows you to display the usage information for NightSim and then exit.

-v |**--version**

Allows you to display the version and copyright information for NightSim and then exit.

-offline

Creates a NightSim window initially in OffLine mode. See "Online/Offline Operation" on page 4-6 for more information.

-s [*targetname*,]*key*
--scheduler=[*targetname*,]*key*

Configures a scheduler associated with an FBS with the key *key* on the system *targetname*. If *targetname* is not specified, the window will be associated with an FBS on the local system where **nsim** is invoked.

key is an integer value that identifies the scheduler.

-t *targetname*
--target=*targetname*

Configures a new scheduler on system *targetname*.

-*Xoption*

Specifies any standard X Toolkit command-line option.

[**-f**] *file*

Allows you to specify the name of a file that contains configuration and scheduling data for a selected *frequency-based scheduler.*

To save the configuration and scheduling data entered in a particular Scheduler page to a file, select the Scheduler->Save Config As menu item. See Save Config As on page 3-4 for details.

You may also use a text editor to create or modify a configuration file, following the format as described in "Configuration Files" on page C-1.

Options can be specified in any order.

You may also invoke NightSim without specifying any options. Doing so allows you to display an inactive window. An inactive window is one that is not associated with a scheduler that exists on the system.

The steps for invoking NightSim are as follows:

1. Log in to your system.
2. Ensure that the value of your `DISPLAY` variable is set to the name of your X server (or use the **`-display`** command-line option to the NightSim command). If using a remote target system, please ensure that your display name is reachable from the target system as well as the GUI host.
3. Type **`nsim`** and any desired options after the system command prompt, and press the Enter key.

The Menu Bar

The menu bar provides access to configuration settings, tools, scheduler operations, obtaining help, and controlling which areas are displayed in the window.

The menu bar provides the following menus:

- File
- View
- Scheduler
- Process
- Monitor
- Tools
- Help

Each menu is described in the sections that follow.

Note that an icon appearing at the left of a menu item indicates that the corresponding icon on the toolbar can also be used to perform that function.

File Menu

Mnemonic: Alt+F

The File menu allows you to load or save configuration data and contains the means to exit NightSim.

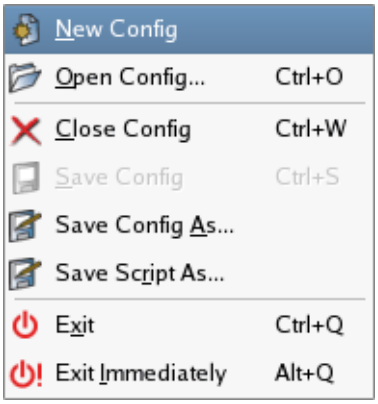


Figure 3-1. File Menu

The following paragraphs describe the options on the File menu in more detail.

New Config

Mnemonic: N

This option clears all information from the current window, resets the various areas to blank or default values and places the window in the inactive state.

Open Config...

Mnemonic: O
Accelerator: Ctrl+O

This option allows you to open a scheduler configuration file that you have previously saved.

To assist you in specifying a configuration file to be opened, NightSim displays a file selection dialog. After making a selection, you may open the selected file, search for another file or cancel the operation. When you open the selected file, the Scheduler window is associated with that file and displays the configuration and scheduling data contained in the file.

NightSim will open the configuration file in Offline mode. Once the file has been read in, NightSim will make a transition to Online mode unless the **`-offline`** option was used on the command line. When this transition occurs, the new scheduler will be verified. See "Online/Offline Operation" on page 4-6 for more information.

Close Config

Mnemonic: C
Accelerator: Ctrl+W

This option clears all information from the current window, resets the various areas to blank or default values and places the window in the inactive state.

Save Config

Mnemonic: S
Accelerator: Ctrl+S

This option allows you to save the configuration and scheduling data entered in the current window in the file that is associated with the window. If the configuration file name has not yet been specified, this option is disabled; you should use the Save Config As... option. See "Configuration Files" on page C-1 for a description of the format of this file as well as "Configuration File Example" on page C-2 for an example.

Save Config As...

Mnemonic: A

This option allows you to specify the name of the file in which you wish the configuration and scheduling data entered in the current window to be saved.

When you select this option, NightSim displays a file selection dialog.You may save the current configuration and scheduling data in a selected file, name a new config file, search for another file or cancel the operation.

See "Configuration Files" on page C-1 for a description of the format of this file as well as "Configuration File Example" on page C-2 for an example.

Save Script As...

Mnemonic: R

This option allows you to save the current configuration as a shell script containing **`rtcp`** commands. The script may be run from any host in your network as long as it has **`rsh(1)`** access to all the targets, as well as the *interrupt host* (see "Remote Shell Access" on page 2-1). This script can then be used to recreate your scheduler configuration without the added overhead of NightSim, allowing your application to start up and run more quickly. This can be extremely useful when dealing with time-critical applications.

The main disadvantage of using this script instead of using NightSim is that you do not get any monitoring capabilities. This may be less important for systems that are

typically used in production environments but can be critical to those systems used in a testing/development environment where the monitoring feedback that NightSim provides can be invaluable.

Another disadvantage is that the script does not support starting your processes under NightView or **`xterm(1)`**. For these features, you must use the NightSim scheduler to start your scheduler.

If a scheduler which makes use of **`xterm`** or NightView is saved as a script, NightSim will ignore these points and schedule the processes directly. If File input or output is used, NightSim will honor those filenames, but if the scheduler uses Standard Input Stream or Standard Output Stream, NightSim will forcibly redirect from or to **`/dev/null`** in order to prevent **`rsh`** from hanging while waiting for input or output.

Exit

Mnemonic: X
Accelerator: Ctrl+Q

This option allows you to exit NightSim. If you have an active scheduler, NightSim displays a dialog verifying that you want to exit, warning that the schedulers will remain active if you exit.

If you confirm that you want to exit, NightSim then checks whether there are changes to the current configuration which have not yet been saved. If so, you will be asked if you wish to save the changes. If you click on the Yes button to save the changes, NightSim saves the current configuration and then exits. If the current configuration is unnamed, a file selection dialog will be displayed. If you click on No, NightSim exits without saving the current configuration. Clicking on Cancel cancels the exit.

If there are no active schedulers set up and/or running and no outstanding changes to be written to the config file, NightSim exits immediately.

Exit Immediately

Mnemonic: I
Accelerator: Alt+Q

This option exits NightSim immediately without the verification and checks performed by the Exit option.

View Menu

Mnemonic: Alt+V

The View menu allows you to control the tools and areas displayed in the current NightSim session.

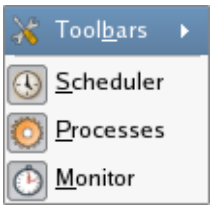


Figure 3-2. View Menu

The following paragraphs describe the options on the View menu in more detail.

Toolbars

Mnemonic: B

This menu item allows control of which tools are displayed in the toolbar. The toolbar is discussed in more detail in the section "The Toolbar" on page 3-11.

Scheduler

Mnemonic: S

This menu item toggles display of the Scheduler area. The Scheduler area is detailed in "Using the Scheduler Area" on page 4-1.

Processes

Mnemonic: P

This menu item toggles display of the Processes area. The Processes area is detailed in the section "Using the Processes Area" on page 4-20.

Monitor

Mnemonic: M

This menu item toggles display of the Monitor area. The Monitor area is detailed in "Using the Monitor Area" on page 5-1.

Scheduler Menu

Mnemonic: Alt+S

The Scheduler menu allows you to control the scheduler.

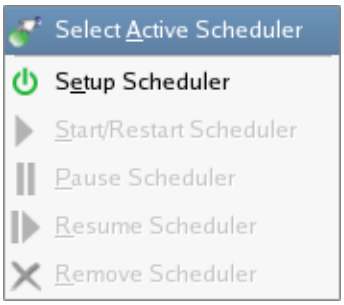


Figure 3-3. Scheduler Menu

This menu is discussed in the section "The Scheduler Menu" on page 4-6.

Process Menu

Mnemonic: Alt+P

The Process menu allows you to define the programs scheduled on a *frequency-based scheduler.*

Figure 3-4. Process Menu

This menu is discussed in the section "The Process Menu" on page 4-21.

Monitor Menu

Mnemonic: Alt+M

The Monitor menu contains general commands related to performance monitoring.

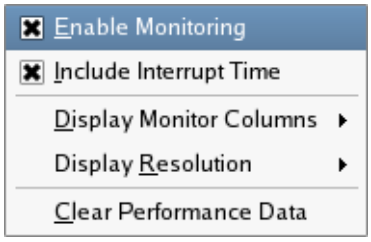


Figure 3-5. Monitor Menu

This menu is discussed in the section "The Monitor Menu" on page 5-8.

Tools Menu

Mnemonic: Alt+T

The Tools menu contains a list of other NightStar tools that can be launched directly from NightSim.

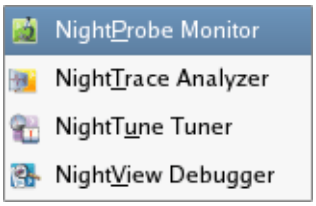


Figure 3-6. Tools Menu

Descriptions of the options on the Tools menu follow.

NightProbe Monitor

Mnemonic: P

Opens the NightProbe Data Monitoring application.

NightProbe is a real-time graphical tool for monitoring, recording, and altering program data within one or more executing programs without significant intrusion. NightProbe can be used in a development environment as a tool for debugging, or in a production environment to create a "control panel" for program input and output.

See the *NightProbe RT User's Guide* for more information.

NightTrace Analyzer

Mnemonic: T

Opens the NightTrace Analyzer.

The NightTrace Analyzer is a graphical tool for analyzing the dynamic behavior of multiprocess and/or multiprocessor user applications and operating system activity. NightTrace allows you to control user and kernel trace collection daemons and can graphically display the interplay between many real-time programs and processes across multiple processors and systems.

See the *NightTrace RT User's Guide* for more information.

NightTune Tuner

Mnemonic: U

Opens the NightTune Performance Tuner.

NightTune is a graphical tool for analyzing the status of the system in terms of processes, interrupts, context switches, interrupt CPU affinity, processor shielding and hyperthreading control as well as network and disk activity. NightTune can adjust the scheduling attributes of individual or groups of processes, including priority, policy, and CPU affinity.

See the *NightTune RT User's Guide* for more information.

NightView Debugger

Mnemonic: V

Opens the NightView Source-Level Debugger.

NightView is a graphical source-level debugging and monitoring tool specifically designed for real-time applications. NightView can monitor, debug, and patch multiple real-time processes running on multiple processors with minimal intrusion.

See the *NightView RT User's Guide* for more information.

Help Menu

Mnemonic: Alt+H

NightSim provides on-line help information through the Help menu. Figure 3-1 presents this menu:

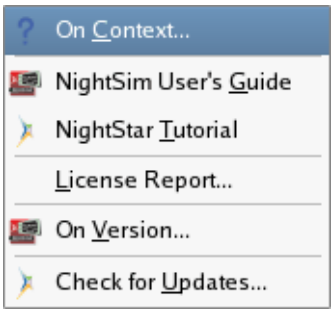


Figure 3-1. Help Menu

The following describe the options on the Help menu:

On Context...

Mnemonic: C

Gives context-sensitive help on the various menu options, dialogs, or other parts of the user interface.

Help for a particular item is obtained by first choosing this menu option, then clicking the mouse pointer on the object for which help is desired (the mouse pointer will become a floating question mark when the On Context menu item is selected).

In addition, context-sensitive help may be obtained for the currently highlighted option by pressing the F1 key. QT-Assistant, NightSim's online help system, will open with the appropriate topic displayed.

NightSim User's Guide

Mnemonic: G

Opens the online version of the *NightSim RT User's Guide* in the online help viewer.

NightStar Tutorial

Mnemonic: T

Opens an online help window containing a tutorial which demonstrates the features of NightSim™, NightProbe™, NightView™, NightTrace™ and NightTune™ in one cohesive example.

License Report...

Mnemonic: L

Provides licensing information.

On Version

Mnemonic: V

Displays a short description of the current version of NightSim.

Check for Updates...

Mnemonic: U

Launches NUU (Network Update Utility) enabling you to update your system. This requires network access to Concurrent's Updates web site. Updates require a login and user ID issued by Concurrent. Refer to **http://redhawk.ccur.com/updates** for complete information.

NightSim also provides help information through the Help buttons that appear in dialog windows. When you click on the Help button in one of these windows, NightSim displays the help information for that window.

The Toolbar

The NightSim toolbar provides icons for commonly used actions.

Figure 3-2. NightSim Toolbar

File Toolbar:

Configures a new NightSim scheduler

Opens an existing NightSim configuration. See Open Config on page 3-3 for details.

Saves the current NightSim scheduler configuration. See Save Config on page 3-4 for details.

Scheduler Toolbar:

Selects a scheduler that is active on the target system. See Select Active Scheduler for details.

Sets up the configured scheduler. See Setup Scheduler on page 4-9 for details.

Toggles between start and restart a scheduler, resetting the frame and cycle counters to zero. See Start/Restart Scheduler on page 4-10 for details.

Toggles between pause and resume the scheduler. See Pause Scheduler on page 4-11 and Resume Scheduler for details.

Removes a configured scheduler. See Remove Scheduler on page 4-12 for details

Online Toolbar:

Toggles scheduler online/offline state. See the section "Online/Offline Operation" on page 4-6 for details

Using Accelerators

A set of keyboard accelerators is defined for use within NightSim to provide quick access to certain functions and menu items. Accelerators associated with menu items allow you to access those items whether or not the menu is posted. Table 3-1 contains a list of the accelerators and the resulting actions; where applicable, it indicates the menu items for which the accelerators provide shortcuts. Note that you can define additional accelerators through the use of X resources (refer to the **`X(1)`** system manual page).

Table 3-1. Accelerators

Accelerator	Menu Item	Action
Ctrl+W	File->Close Config	Closes the open configuration file.
Ctrl+Q	File->Exit	Exits NightSim with verification if schedulers are active or the current configuration is not yet saved
Alt+Q	File->Exit Immediately	Exits NightSim immediately without any checks or verification
Ctrl+O	File->Open Config	In the current window, opens a scheduler configuration file that you have previously saved
Ctrl+S	File->Save Config	Saves the configuration and scheduling data entered in the current window in the file that is associated with the window
F1		Displays context help for the component that currently has the focus

Exiting NightSim

You may exit NightSim by using one of the following techniques:

- Select the File->Exit menu item or press Ctrl+Q. If schedulers are active or the current configuration is not yet saved, you will be asked to confirm that you wish to exit.
- Select the File->Exit Immediately menu item or press Alt+Q. NightSim will exit without checking for active schedulers or unsaved configurations and does not ask for verification.

4
Using the Scheduler Area

This chapter describes the *frequency-based scheduler* and explains how scheduler frequency is defined and how processes are scheduled. It introduces the components of the Scheduler area of the NightSim window and describes the use of each component.

Understanding the Frequency-Based Scheduler

The *frequency-based scheduler* is a task synchronization mechanism that allows you to run processes at frequencies that you specify. Frequencies can be based on high-resolution clocks, an external interrupt source, or completion of a cycle. The frequency-based scheduler provides a mechanism for initiating processes at the specified frequency.

The frequency–based scheduler provides you with the ability to:

- Define *FBS* frequency in terms of the duration of a *minor cycle* and the number of minor cycles per *major frame*
- Specify the scheduling parameters with which processes are scheduled
- Detect *frame overruns* for all FBS–scheduled processes
- View the status of FBS–scheduled processes
- Remove one or all FBS–scheduled processes from a scheduler
- Reschedule an FBS–scheduled process
- Start, stop, and resume scheduling on a frequency–based scheduler
- Connect an interrupt source to and disconnect it from a frequency–based scheduler
- Control use of the real–time clock device as the interrupt source for a frequency–based scheduler
- Configure up to 100 frequency–based schedulers system–wide in a single processor or multiprocessor environment
- Use both frequency–based scheduling and static priority scheduling simultaneously

In addition, NightSim provides the following additional capabilities for your convenience:

- Specify input and output files for scheduled processes
- Specify a working directory for scheduled processes
- Schedule processes under **`xterm(1)`** terminal windows, providing the option of interactive behavior

- Schedule processes under the NightView debugger

NOTE

Some of these additional features are not available for schedulers started via a saved script.

How Is Scheduler Frequency Defined?

You configure a *frequency–based scheduler*, in part, by defining the number of *minor cycles* that compose a *major frame*. Minor cycles and major frames have associated with them a duration of time that you can define by using a interrupt source for the scheduler. The interrupt source can be the end of a minor cycle, a real–time clock, an edge–triggered interrupt, or a user–supplied device.

If you use *end-of-cycle scheduling*, scheduling is triggered when the last process that is scheduled during the current minor cycle of the current major frame completes its processing.

If you use a real–time clock as the interrupt source, you can specify the period in microseconds or the frequency in hertz.

How Are Processes Scheduled?

You schedule processes to run at a certain frequency by specifying the first *minor cycle* in which the process is to be wakened in each *major frame* (called the *starting base cycle*) and the frequency with which it is to be wakened (called the period).

If, for example, you schedule "Process 1" with a starting base cycle of zero and a *period* of two, the process will be wakened once every two minor cycles, starting with the first minor cycle in the frame.

If you schedule "Process 2" with a starting base cycle of one and a period of four, that process will be wakened once every four minor cycles, starting with the second minor cycle in the frame.

If you then schedule "Process 3" with a starting base cycle of two and a period of two, that process will be wakened once every two minor cycles, starting with the third minor cycle in the frame.

On a *frequency–based scheduler* configured with 100 minor cycles per major frame, these processes will be wakened as illustrated in Table 4-1.

Table 4-1. Process Scheduling

Minor Cycle	Processes Wakened
0	Process 1
1	Process 2
2	Process 1, Process 3
3	
4	Process 1, Process 3
5	Process 2
...	
97	Process 2
98	Process 1, Process 3
99	

The maximum frequency with which you can schedule a process is once per minor cycle (a period of one); the minimum frequency is once per major frame (in the case of the example, a period of 100).

A process runs until it calls an *FBS* library routine that causes it to sleep until the frequency–based scheduler wakes it again. The frequency–based scheduler wakes those sleeping processes that are scheduled to be wakened in the current minor cycle of the current major frame and repeats the process for each minor cycle in the current frame. It continues to repeat the entire process on every major frame until the scheduler is disabled. A scheduler configured with 100 minor cycles per major frame, a minor cycle duration of 10,000 microseconds (0.01 second), and a major frame duration of one second wakes processes as illustrated in Table 4-2.

Table 4-2. Scheduler Operation

Major Frame	Time (sec.)	Minor Cycle	Processes Wakened
0	0	0	Process 1
	0.01	1	Process 2
	0.02	2	Process 1, Process 3
	...		
	0.97	97	Process 2
	0.98	98	Process 1, Process 3
	0.99	99	

Table 4-2. Scheduler Operation (Cont.)

Major Frame	Time (sec.)	Minor Cycle	Processes Wakened
1	1.00	0	Process 1
	1.01	1	Process 2
	1.02	2	Process 1, Process 3
	...		
	1.97	97	Process 2
	1.98	98	Process 1, Process 3
	1.99	99	
	...		
n	n.00	0	Process 1
	n.01	1	Process 2
	n.02	2	Process 1, Process 3
	..		
	n.97	97	Process 2
	n.98	98	Process 1, Process 3
	n.99	99	

As illustrated in Table 4-2:

- When the current major frame is zero and the current minor cycle is zero, the scheduler wakes "Process 1."
- After 0.01 second, it wakes "Process 2"; after 0.02 second, it wakes "Process 1" and "Process 3"; and so on.
- At one second, when the current major frame becomes one, the current minor cycle becomes zero again, and the scheduler wakes "Process 1."
- After .01 second, it wakes "Process 2."
- After .02 second, it wakes "Process 1" and "Process 3"; and so on.
- The scheduler continues repeating this process for as long as it is enabled.

It is important to note that a process may not always run at the frequency that you have specified. A *frame overrun* occurs when a scheduled process does not finish its processing before it is scheduled to run again. Overrun counts are maintained for each process and for each scheduler. You can obtain information on overruns by querying a process in the Monitor area (see "Using the Monitor Area" on page 5-1). When you schedule a process on a selected scheduler, you have the option of ensuring that the scheduler is stopped when that process causes a frame overrun.

For additional information on the *frequency-based scheduler*, refer to the *RedHawk Frequency-Based Scheduler (FBS) User's Guide (0898005)*.

Programming Requirements

Programs scheduled to run on a frequency-based scheduler have certain requirements:

- The program must call **`fbswait(3)`**.
- The program must complete its work unit and return to **`fbswait`** before its next scheduled cycle.
- The program will be invoked with one argument. `argc` will be 2 and `argv[1]` will be 0 or a string representation of the parameter specified on the Edit Process dialog's FBS Schedule page (see "FBS Schedule" on page 4-31).
- Unscheduled processes invoked via the FBS will also be invoked with a single parameter argument. It is not possible to suppress this argument or to pass additional parameters. This is because **`sched_pgmadd(3)`** is used to create the process instead of **`exec(1)`**.

When scheduling a program on the FBS under the control of an X terminal or to schedule processes under NightView, there is an important step your program must take to be able to perform interactive input. The **`sched_pgmadd`** function starts the user process with the X terminal as the controlling terminal, but the controlling terminal's process group is the parent process. This has the net effect of making the scheduled process a 'background' process in terms of I/O and job control behavior. This means that if your program tried to read from `stdin`, the process will receive signal SIGTTIN, the **`read()`** call will be interrupted, and will fail. To permit interactive input to your scheduled process, you must update the process group of the controlling terminal. To do this, simply add this code, or its equivalent in the programming language of your program:

```
if (isatty(0)) {
    if (tcgetpgrp (0) != getpgid(getpid()))
        tcsetpgrp (0, getpgid(getpid()));
```

Once this is done, you will be able to read from standard input.

In addition, you need to block SIGTTOU with the following:

```
sigset_t bset;
sigset_t oset;

sigemptyset (&bset);
sigemptyset (&oset);
sigaddset (&bset, SIGTTOU);
sigprocmask (SIG_BLOCK, &bset, &oset);
if (tcsetpgrp (0, pgid) < 0)
    perror ("tcsetpgrp");
sigprocmask (SIG_SETMASK, &oset, &oset);
```

Refer to the *RedHawk Linux Frequency-Based Scheduler User's Guide*, publication number 0898005, for more information about FBS programming.

Online/Offline Operation

NightSim provides two modes of operation for a scheduler: Online and Offline. You can toggle the current state by using the Server Online/Offline dropdown on the online toolbar (see "The Toolbar" on page 3-11).

Figure 4-1. Online Toolbar

By placing a NightSim scheduler into an Offline state, immediate checking of the information entered in either the Scheduler area or the Processes area can be avoided. When in Offline mode, the NightSim scheduler will not communicate with any servers on any interrupt host or target system of the scheduler.

This is useful when creating a configuration for future use that deals with target systems which may not currently exist or may not be currently accessible. The values entered in a scheduler while it is Offline will be checked when the mode is changed to Online.

There are two ways to request a transition to Online mode:

1. Select Online from the Online toolbar.
2. Press the Set Up button in the Scheduler Control area.

In each of these cases, when going from Offline to Online, the information entered in the Scheduler Definition area and the Edit Process dialog is examined for correctness given the current state of the interrupt host and target systems. NightSim servers are started if necessary, and all scheduled process attributes are checked for correctness.

The Scheduler Menu

Scheduler control can be performed using selections from the Scheduler Menu of the NightSim window menu bar (see "The Menu Bar" on page 3-2). It can also be performed using the Scheduler Control area (see "Using the Scheduler Control Area" on page 4-12) and tools on the NightSim toolbar (see "The Toolbar" on page 3-11).

The following figure illustrates NightSim's Scheduler menu.

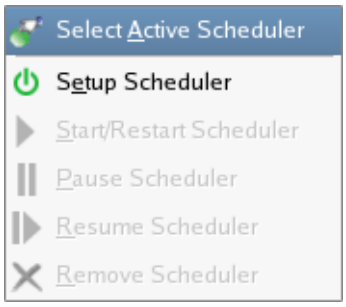


Figure 4-2. Scheduler Menu

The following paragraphs describe the options on the Scheduler menu in more detail.

Select Active Scheduler

Mnemonic: A

This option allows you to associate the current NightSim window with a *frequency-based scheduler* active on a given *target system*. When you select this option, NightSim displays Figure 4-3.

NOTE

NightSim needs to communicate with a target system in order to determine which schedulers are active. This will occur even if the existing window is Offline. This is to be expected since the existing scheduler will be abandoned when an active scheduler is selected, and because an active scheduler is necessarily Online (see “Online/Offline Operation” on page 4-6).

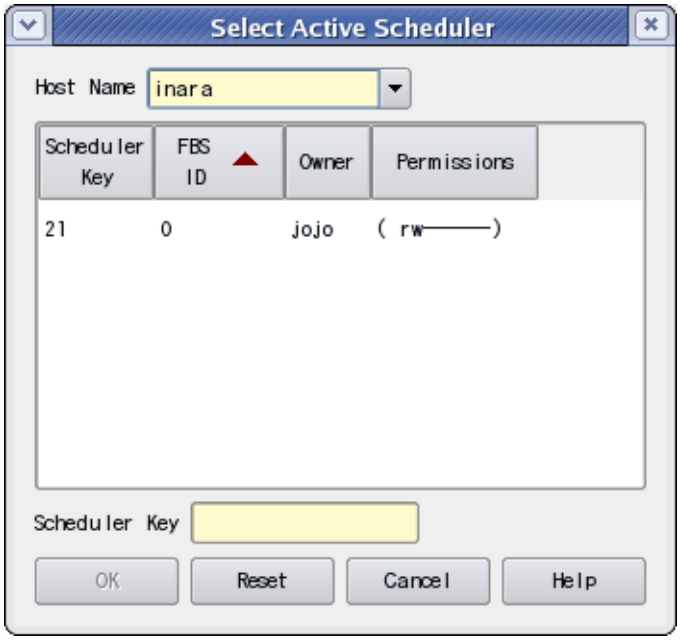


Figure 4-3. Select Active Scheduler Dialog

This dialog provides a list of the *active schedulers* for the system designated by the Host Name field. The Host Name defaults to your local host and the list contains the active schedulers running on that particular system. For instance, Figure 4-3 lists the two schedulers currently running on the host `raptor`.

You may change the Host Name either by using the drop-down list or by typing the name of the *target system* into the text field provided. Entering an item in this field will populate the list with the *active schedulers* on the specified target.

You may select a scheduler from the list that the dialog displays, or you may use the Scheduler Key text field to enter the numeric *key* that is associated with the desired scheduler. The key can be any positive integer value. Whether you select from the list or enter a key, the selected scheduler must previously have been configured, and you must have at least read permission for it; otherwise, NightSim reports an error.

After making your selections, click OK to apply the selections and close the dialog, Reset to reset original values, Cancel to discard selections and close the dialog, or Help to display help.

If you successfully select an existing scheduler, the NightSim window becomes an active window associated with that scheduler. Information related to the selected scheduler replaces information that the window contained prior to the operation.

The Select Active Scheduler option is equivalent to using the  icon on the scheduler toolbar (see "The Toolbar" on page 3-11).

Setup Scheduler

Mnemonic: E

This menu item is used to set up the configured scheduler. It is enabled only when the window is in the *inactive* state.

When you select Setup Scheduler, NightSim carries out the following actions:

- Transitions the NightSim Scheduler to an Online mode of operation, if the NightSim Scheduler is operating in an Offline mode (see “Online/Offline Operation” on page 4-6). The state of the target on which processes are scheduled, the existence and execute permissions of each scheduled program, and the existence and read permissions for any input files required are verified.
- Determines whether there is an existing scheduler that is associated with the same *key* as that entered in the current window.

  If an active scheduler with the same scheduler key as the key you selected already exists on the target system to which you assigned processes, NightSim displays the dialog that Figure 4-4 presents.

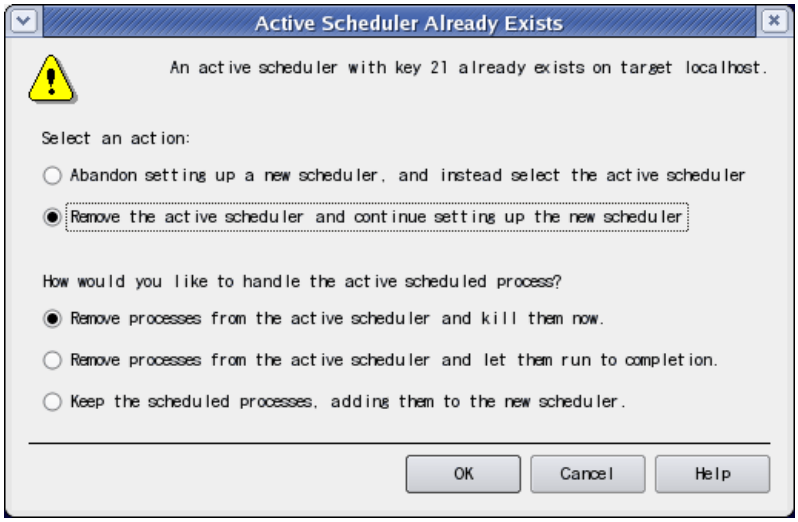


Figure 4-4. Scheduler Exists Dialog

You have two main choices:

- Work with the existing scheduler, in which case the active scheduler’s configuration will appear in the Scheduler Definition area in place of the current configuration.

or

- Replace the existing scheduler with a new scheduler based on the configuration in the window.

If you decide to replace the existing scheduler, you then need to decide what to do with the processes running under the old scheduler.

After making your selections, click on the OK button. Click on Cancel to cancel the selection and close the dialog.

You may also click on the Cancel button to abort this action.

- Creates a scheduler that is configured according to the parameters specified in the Scheduler Definition area of the window (see "Using the Scheduler Definition Area" on page 4-16 for details).
- Schedules the processes that are listed in the Processes Area of the window and starts them running (see "Using the Processes Area" on page 4-20 for details). Programs will run up to the first **`fbswait`** system call.
- Attaches the interrupt source to the scheduler.
- Enables the Start function. Initially, the simulation is stopped.
- Disables the Set up function and enables the Pause and Remove functions.
- Places the window in the *active* state and changes the state icon accordingly.

The Setup Scheduler menu option is equivalent to using both the  icon on the scheduler toolbar (see "The Toolbar" on page 3-11) and the Set up button in the Scheduler Control area of the window (see "Using the Scheduler Control Area" on page 4-12).

Start/Restart Scheduler

Mnemonic: S

This option starts or restarts a stopped simulation. It is enabled only when the simulation is stopped.

When you select Start/Restart Scheduler, NightSim carries out the following actions:

- Attaches the interrupt source to the scheduler if not already attached or if the interrupt source has been changed.
- If a real-time clock is being used as the interrupt source, sets the clock period in accordance with the value entered in the Clock Period/Freq field in the Scheduler Definition area (see "Clock Period/Freq" on page 4-19).
- Starts the simulation with the values of the *minor cycle*, *major frame*, and *overrun* counts set to zero.

The Start/Restart Scheduler option is equivalent to using both the  icon on the scheduler toolbar (see "The Toolbar" on page 3-11) and the Start button in the Scheduler Control area of the window (see "Using the Scheduler Control Area" on page 4-12).

Pause Scheduler

Mnemonic: P

This option stops the currently running simulation. It is enabled only when the simulation is running.

When you select Pause Scheduler, NightSim stops the simulation. When the simulation is stopped, the interrupt source is not detached from the scheduler, but the Interrupt source and Clock Period/Freq fields in the Scheduler Definition area are enabled. You may wish to change the interrupt source, and, if the interrupt source is a real-time clock, you may wish to change the clock *period*.

NOTE

If you change the interrupt source, the previously specified interrupt source is detached from the scheduler when you select the Start/Restart or Resume functions to restart or resume the simulation.

While the simulation is stopped, you may also wish to change one or more of the parameters with which you have scheduled a process on the scheduler.

Although you can also change the parameters with which a process has been scheduled while the scheduler is running, it is not recommended that you do so. Rescheduling requires processor time. As a result, other processes that are currently scheduled on the *frequency-based scheduler* may experience *frame overruns*. If you have indicated as such, the scheduler will halt on a frame overrun but can be resumed.

The Pause Scheduler option is equivalent to using both the  icon on the scheduler toolbar, which toggles between pause and resume, (see “The Toolbar” on page 3-11) and the Pause button in the Scheduler Control area of the window (see “Using the Scheduler Control Area” on page 4-12).

Resume Scheduler

Mnemonic: R

This option starts the stopped simulation from the point at which it was stopped. It is only enabled when the simulation has been running but is currently stopped.

When you select Resume Scheduler, NightSim restarts the simulation with the values of the *minor cycle*, *major frame*, and *overrun* counts the same as they were when you stopped the simulation.

The Resume Scheduler option is equivalent to using both the  icon on the scheduler toolbar, which toggles between pause and resume, (see “The Toolbar” on page 3-11) and the Resume button in the Scheduler Control area of the window (see “Using the Scheduler Control Area” on page 4-12).

Remove Scheduler

Mnemonic: R

This option allows you to remove the active scheduler from the system. It is enabled only when the window is in the *active* state.

When you select Remove Scheduler, NightSim carries out the following actions:

- Displays a dialog verifying that NightSim should kill the processes running on the scheduler. You may press Yes to kill the processes that are currently scheduled on the scheduler, No to remove the processes from the scheduler but allow them to continue executing, or Cancel to abort the action.
- Stops the simulation if it is running.
- If a real-time clock is being used as the interrupt source, and the scheduler is running, stops the clock.
- Detaches the interrupt source from the scheduler or disables *end-of-cycle scheduling.*
- Removes all processes that are currently scheduled on the scheduler from it and either terminates them or allows them to continue executing in accordance with the selections made in response to the dialog presented.
- Removes the scheduler from the system.
- Disables the Start, Pause, and Resume functions.
- Enables the Set up function and disables the Remove function.
- Places the scheduler in the *inactive* state and changes the state icon accordingly.

The Remove Scheduler option is equivalent to using both the  icon on the scheduler toolbar (see "The Toolbar" on page 3-11) and the Remove button in the Scheduler Control Area of the window (see "Using the Scheduler Control Area" on page 4-12).

Using the Scheduler Control Area

The Scheduler Control area allows you to create and remove a *frequency-based scheduler* and start, stop and resume a simulation with push buttons that you use to perform these functions. Status of the simulation run is also provided.

The Scheduler Control area is shown in Figure 4-5.

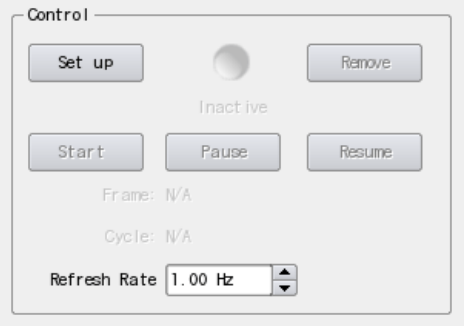


Figure 4-5. Scheduler Control Area

Descriptions of the buttons and fields on this page follow:

Set up Button

The Set up button is used to set up the configured scheduler. It is enabled only when the window is in the *inactive* state.

Pressing the Set up button is equivalent to using both the icon on the scheduler toolbar (see "The Toolbar" on page 3-11) and the Setup Scheduler selection of the Scheduler menu. See Setup Scheduler on page 4-9 for a complete description of the operations that are performed.

Remove Button

The Remove button allows you to remove the active scheduler from the system. It is enabled only when the window is in the *active* state.

Pressing the Remove button is equivalent to using both the icon on the scheduler toolbar (see "The Toolbar" on page 3-11) and the Remove Scheduler selection from the Scheduler menu. See Remove Scheduler on page 4-12 for a complete description of the operations that are performed.

Start Button

The Start button starts or restarts a stopped simulation. It is enabled only when the simulation is stopped.

Pressing the Start button is equivalent to using both the icon on the scheduler toolbar (see "The Toolbar" on page 3-11) and the Start/Restart Scheduler selection in the Scheduler menu. See Start/Restart Scheduler on page 4-10 for a complete description of the operations that are performed.

If the simulation has been paused, the Start button label changes to Start over and when clicked, the simulation is restarted with all values starting at zero.

Pause Button

The Pause button stops the currently running simulation. It is enabled only when the simulation is running.

Pressing the Pause button is equivalent to using both the  icon on the scheduler toolbar, which toggles between pause and resume, (see "The Toolbar" on page 3-11) and the Pause Scheduler menu item in the Scheduler menu. See Pause Scheduler on page 4-11 for a complete description of the operations that are performed.

Resume Button

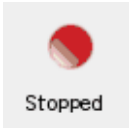
This option starts the stopped simulation from the point at which it was stopped. It is enabled only when the simulation has been running but is currently stopped.

When you select Resume, NightSim restarts the simulation with the values of the *minor cycle*, *major frame*, and *overrun* counts the same as they were when you stopped the simulation.

Pressing the Resume button is equivalent to using both the  icon on the scheduler toolbar, which toggles between pause and resume, (see "The Toolbar" on page 3-11) and the Resume Scheduler menu item in the Scheduler menu. See Resume Scheduler on page 4-11 for a complete description of the operations that are performed.

State of the Simulation

An icon is provided on the Scheduler Control area that indicates whether the simulation is stopped, running or inactive.

	When the simulation is stopped, NightSim displays a red orb.
	When the simulation is running, NightSim displays a green orb.
	When the window is inactive, the state of the simulation area is grayed.

Frame

This field shows the number of the current major frame for the simulation running on the scheduler.

Cycle

This field shows the number of the current minor cycle for the simulation running on the scheduler.

Refresh Rate

This field allows you to specify the frequency with which the frame and cycle counts are to be updated. Enter the number of seconds to occur between updates. The minimum value that you may specify is .01 second; the maximum is 100 seconds. The default value is 1.0 second.

Using the Scheduler Definition Area

The Scheduler Definition area allows you to configure a *frequency-based scheduler*, specify maximum values for the scheduler, and specify the permissions that are to be associated with it.

The Scheduler Definition area is shown in Figure 4-6.

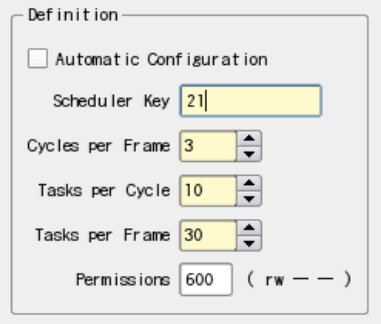


Figure 4-6. Scheduler Definition Area

Descriptions of the fields contained in the Scheduler Definition area follow:

Automatic Configuration

If this field is checked, NightSim chooses reasonable scheduler dimensions based on the start cycle and period values of the processes scheduled. A frame size is chosen such that each process will be scheduled at least twice within a frame.

If this field is unchecked, the user is responsible for choosing appropriate scheduler dimensions.

As a convenience, NightSim chooses scheduler dimensions with enough extra space to allow one **`/idle`** process per CPU and at least one additional process to be added after the scheduler is set up.

Scheduler Key

This field allows you to specify a *key* for the frequency–based scheduler that you wish to create. The key is a user–chosen numeric identifier with which the scheduler will be associated. It can be any positive integer value. Note that the number of schedulers that can be configured at one time cannot exceed the value of FBSMNI, which is the maximum number of frequency–based schedulers permitted on your system (see "Establishing the NightSim Environment" on page 2-1 for configuration information).

Cycles per Frame

This field allows you to specify the number of cycles that compose a frame on the specified scheduler.

Tasks per Cycle

This field allows you to specify the maximum number of processes that can be scheduled to execute during one cycle.

Tasks per Frame

This field allows you to specify the number of scheduled tasks that may be initiated in a single frame of Cycles per Frame. Every time a process (or **`/idle`**) is scheduled to run within a frame constitutes a single task.

Permissions

This field allows you to specify the permissions required for operations related to the specified scheduler. Enter three octal digits—the first indicating permissions granted to the owner, the second indicating those granted to the group, and the third indicating those granted to other users. The octal method for setting permissions associated with a scheduler is the same as that used for specifying *mode* with the **`chmod`** command (for assistance in using this method, see the system manual page for **`chmod(1)`**). The default, **600,** grants read and alter (write) permission to the owner only.

Using the Scheduler Interrupt Area

The Scheduler Interrupt area allows you to select the interrupt host and the interrupt source that is to be used for the scheduler.

The Scheduler Interrupt area is shown in Figure 4-7.

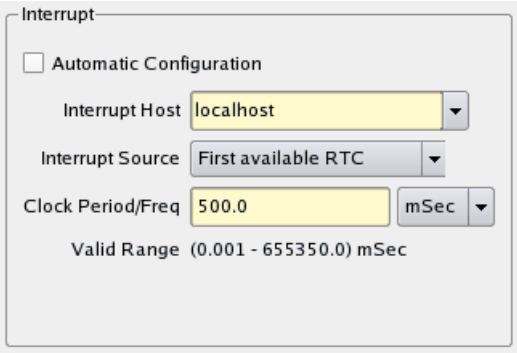


Figure 4-7. Scheduler Interrupt Area

Descriptions of the fields contained in the Scheduler Interrupt area follow:

Automatic Configuration

If this field is checked and an RCIM is installed on the Interrupt Host system, NightSim attempts to select the first real-time clock on the target system which is not already attached to an active scheduler. Once a clock is selected, it will be the preferred clock if the Scheduler is removed and restarted.

NightSim uses 500 milliseconds as the default clock period for the real time clock.

If no RCIM is present, then NightSim selects end-of-cycle scheduling.

Interrupt Host

This field allows you to select the *interrupt host* of the interrupt source for the scheduler. You may enter the name of the host directly in the entry field or choose from the supplied list.

This list contains possible interrupt hosts. NightSim maintains a list in **`~/.NightStarHosts`** of all hosts you ever used as an interrupt host. Once selected, the Interrupt Source menu is repopulated with the set of devices available on that host.

On a NightSim Host which also has the FBS installed, this field will be initialized to the current host name. If the current host is not a valid target system, this field will initially be blank and the user must provide an Interrupt Host name.

NOTE

The interrupt host cannot be changed while the scheduler is active; you must first remove the scheduler.

Interrupt Source

This field allows you to select the interrupt source for the scheduler. The interrupt source can be the end of a *minor cycle*, a real-time clock, an edge-triggered interrupt, or a user-supplied device. Figure 4-8 presents an example of the dropdown menu.

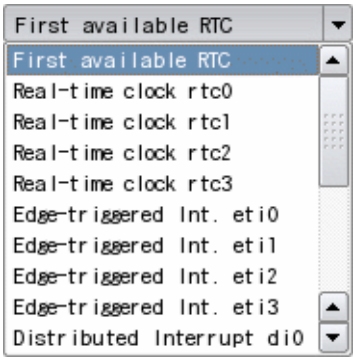


Figure 4-8. Interrupt Source Dropdown Menu

This list contains the set of devices available on the *interrupt host*. Note that the options that NightSim displays vary according to the devices that are configured in the system designated as the interrupt host and may change if the interrupt host is changed. This list will be initially populated with the devices on the default interrupt host.

If you select the End of processing cycle option from the Interrupt Source dropdown menu, scheduling is triggered when the last process that is scheduled during the current *minor cycle* of the current *major frame* completes its processing.

If you select a real-time clock, the Clock Period/Freq text field is enabled so that you can specify the information needed to set the clock.

If you select the Other… option, NightSim displays the dialog shown in Figure 4-9. The text input area in this dialog allows you to specify the name of a user-supplied device that you wish to use as the interrupt source. The Device Pathname must be the name of a device file on the currently selected end-of-cycle_scheduling and that is supported by the RedHawk kernel as a FBS timing device. Currently, the only devices supported are those associated with an RCIM.

Figure 4-9. Other Device Dialog

Clock Period/Freq

If you select a real-time clock as the interrupt source for the scheduler, this field allows you to specify a value to be used in setting the clock. Otherwise, this field is disabled.

You may specify the period in microseconds, milliseconds, hertz, or kilohertz by clicking on the dropdown and choosing the desired option. Figure 4-10 presents the options from which you may choose.

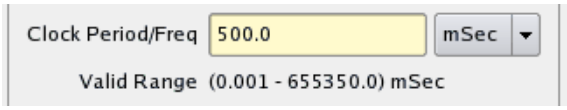


Figure 4-10. Clock Period/Freq Dropdown

Valid Range

The Valid Range field shows the minimum and maximum values that are permitted for the clock period specified in the Clock Period/Freq field. The range changes according to the option you choose.

The valid range of the clock period is determined by multiplying the minimum and maximum clock counts, 1 and 65535, by the clock resolution of 10 microseconds.

Using the Processes Area

The Processes area allows you to:

- add programs to a *frequency-based scheduler*
- remove *FBS*-scheduled processes from a scheduler
- reschedule FBS-scheduled processes
- view scheduling data for current FBS-scheduled processes
- select the processes for which scheduling data are to be displayed

It is important to note that the Processes area, like the other areas, can be "torn off" from the NightSim window to reside in its own window separate from NightSim by clicking on the dashed line at the top of the Area and dragging to another location on the desktop.

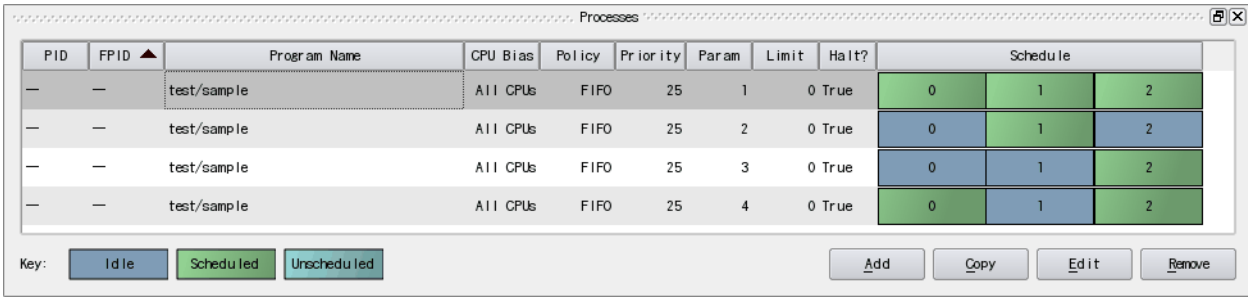


Figure 4-11. Processes Area

The Processes area contains a scrolling list of the processes that are currently scheduled on a *frequency-based scheduler* and the parameters with which they have been scheduled (CPU bias, scheduling policy and priority, starting *minor cycle*, and so on). You can select the process or processes that you wish to reschedule or delete by clicking on the respective item(s) in this list.

In addition, the Processes area includes a graphical representation of how the program will be scheduled within each frame. The labels show the number of the *minor cycle*. The colors indicate the current condition of the process: idle, scheduled or unscheduled as represented in the Key.

Double clicking on certain parameter fields in the Processes area displays the Edit Process dialog that allows you to modify the values in those fields. See the section "Edit Process" on page 4-26 for a description of the Edit Process dialog.

Columns in the Processes area can be resized or hidden using selections from the context menu. The context menu is displayed by right-clicking in the Processes area or on a specific column heading.

The Processes area contains a row of process control buttons below the list of FBS-scheduled processes. These controls (with the exception of Refresh) are also provided by the Process Menu of the NightSim window menu bar and by a context menu that is accessible by right-clicking within the Process area. These controls are explained under the section "The Process Menu" on page 4-21. The Refresh button refreshes the Processes area.

The Process Menu

Processes to be scheduled on a frequency-based scheduler can be selected from the Process Menu of the NightSim window menu bar. They can also be selected using the buttons at the bottom of the Processes area and from the context menu that is accessed by right-clicking within the Process area.

The following figure illustrates NightSim's Process menu.

Figure 4-12. Process Menu

The following paragraphs describe these options in more detail.

Add

Mnemonic: A
Accelerator: Ctrl+N

This selection allows you to add a program to a frequency-based scheduler or measure idle time on a CPU or reschedule an FBS-scheduled process. When you make this selection, NightSim displays the Add Process dialog. Note that if you select a process from the list of scheduled processes and then select Add, the dialog contains the scheduling parameters for the selected process. See the section "Add Process" on page 4-24 for a description of the Add Process dialog.

Copy

Mnemonic: C
Accelerator: Ctrl+C

This selection adds a copy of one or more selected process(es) to the frequency-based scheduler. When you click on a process in the Processes area, this selection is activated.

Edit

Mnemonic: E
Accelerator: Ctrl+E

This selection allows you to edit attributes of FBS-scheduled processes on a scheduler. When you click on a process in the Processes area, this selection is activated. When you make this selection, NightSim displays the Edit Process dialog. The dialog contains the scheduling parameters for the selected process. See the section "Edit Process" on page 4-26 for a description of the Edit Process dialog.

Remove

Mnemonic: R
Accelerator: Ctrl+R

This selection allows you to remove FBS-scheduled processes from a scheduler. All processes that are selected in the list will be removed.

If the scheduler is active when you select Remove, NightSim displays a dialog to confirm killing the selected processes. You may press the Yes button to kill the selected processes after deleting them from the *frequency-based scheduler.* You may also press the No button to delete the processes from the scheduler but allow them to continue executing or press Cancel to abort the action.

If the scheduler is inactive when you select Remove, NightSim displays a dialog to confirm deleting the processes. You may press OK to delete the processes from the scheduler or Cancel to cancel the operation.

Display Process Columns

Mnemonic: D

This option allows you to specify the columns that are to be displayed in the Processes area. When you select this option, NightSim displays the menu that Figure 4-13 presents.

Figure 4-13. Display Process Columns Menu

Descriptions of the options contained in the Display Process Columns menu follow.

Summary Fields

This option specifies the parameters with which the processes have been scheduled.

All Fields

This option specifies all fields available for display in the Processes area.

Custom Fields

This option allows you to select any of the types of values that the Processes area maintains.

When you select this option initially, NightSim displays the dialog that Figure 4-14 presents.

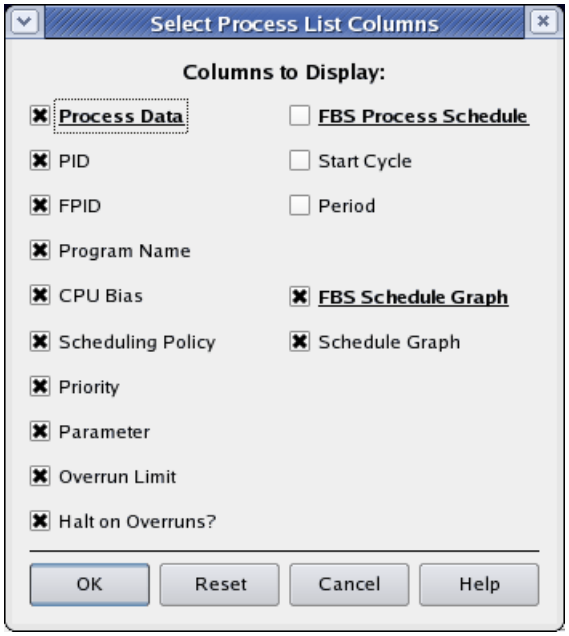


Figure 4-14. Process Columns to Display Dialog

You use the panels of check boxes to select the types of values that you wish to save. Click on one or more boxes as desired.

After you have selected the types of values that you wish to save, select OK to apply the selections and close the dialog box, or you can reset the original settings, cancel the selections, or display help related to the dialog.

Select Fields...

When Custom Fields is enabled and you wish to change the selection of fields, this option displays the Columns to Display dialog so that changes can be made.

Add Process

When the Add option in the Processes area is clicked, the Add Process dialog appears.

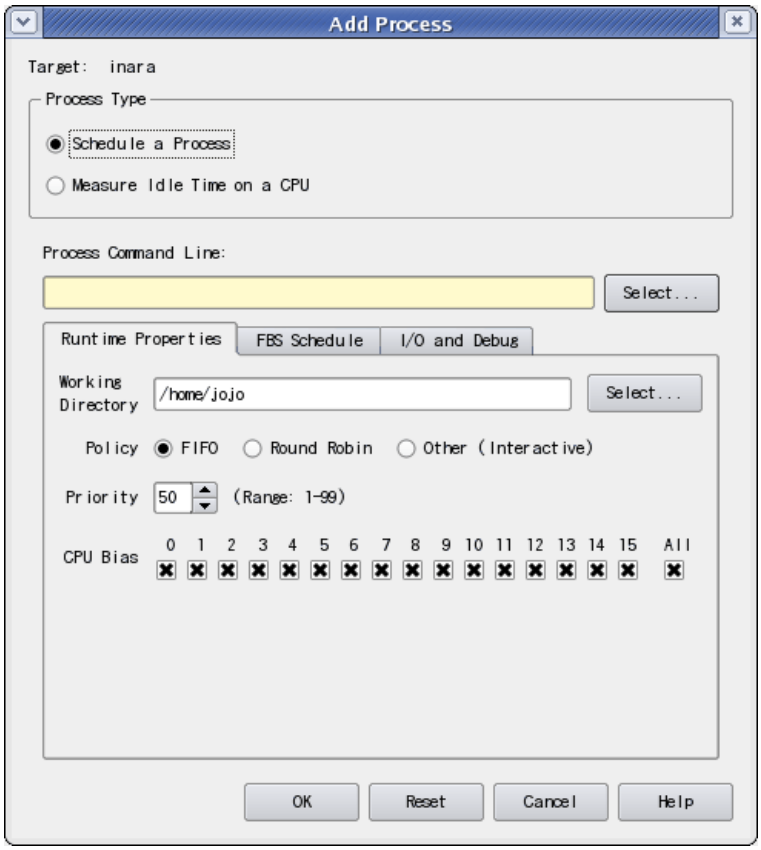


Figure 4-15. Add Process Dialog

The following fields appear at the top of the Add Process dialog:

Target

This item indicates the *target system* on which this process is to be scheduled.

Schedule a Process

When this radio button is filled, the dialog provides information for scheduling and defining a process to be added to the scheduler.

Measure Idle Time on a CPU

When this radio button is filled, the dialog shows the CPU bias of the idle task. When the Add option is selected, the **`/idle`** task is added to the scheduler. See “Monitoring Idle Time” on page 5-4 for more information about monitoring idle time.

Command Line

You may specify the path to the program to be scheduled on the scheduler. Arguments are accepted for all programs except **`/idle`**. The path must be a valid path name on the *target system*. If the target system is not available, or if the scheduler is Offline (see "Online/Offline Operation" on page 4-6), the path must be a valid directory on the target system at the time the program scheduler is created (i.e. when the Set up function is selected).

Select...

You may use the Select button to bring up the Select a program name dialog to specify the directory containing the program you want to configure under the *FBS*. This button works on the path of the program, leaving arguments intact.

The tabs on this dialog are used to display different pages that contain various parameters related to the given process. Refer to the section "Add Process and Edit Process Dialog Pages" on page 4-28 for complete descriptions.

The buttons along the bottom of the dialog control actions to be taken after you have entered the scheduling parameters for a specified program:

OK

Clicking on the OK button makes the specified changes, if any, and closes the dialog.

Reset

Clicking on the Reset button resets the fields to the original values without closing the dialog.

Cancel

Clicking on the Cancel button resets the fields to the original values if changed and closes the dialog.

Help

Clicking on the Help button displays help for the dialog.

Edit Process

The Edit Process dialog allows you to add a program to a *frequency-based scheduler* or reschedule an *FBS*-scheduled process. Note that if you select a process from the list of scheduled processes in the Processes area (see "Using the Processes Area" on page 4-20) and select the Edit option, this dialog will contain the scheduling parameters for the selected process.

The Edit Process dialog is shown in Figure 4-16:

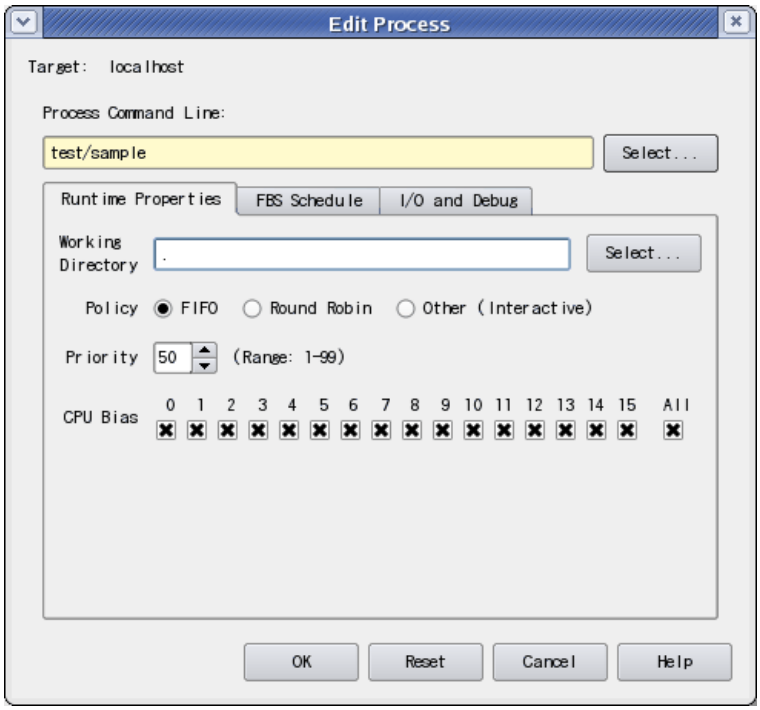


Figure 4-16. Edit Process Dialog

The following fields appear at the top of the Edit Process dialog.

Target

This item indicates the *target system* on which this process is to be scheduled.

Command Line

This field specifies the name of the process that you wish to schedule. Arguments are accepted for all programs except **\idle**. Enter a full or relative pathname (see "Pathname Expansion" on page 4-35). The pathname must be valid on the *target system* to which this process is assigned. If you have selected a process from the list of *FBS*-scheduled processes prior to selecting the Edit option, this field contains the name of the FBS-scheduled process.

NOTE

You cannot change the name of a process while the process is running. You must either stop the scheduler first or add a new process with the desired name.

Select...

You may use the Select button to bring up the Select a program file dialog to choose the program you want to configure under the *FBS*. This button works on the path of the program, leaving arguments intact. Note that NightSim displays files on the selected target system if it is reachable via the network. If the target system is unavailable, NightSim will disable file browsing on the target system. In addition, the Scheduler must be operating in an Online state in order to browse on the target file system (see "Online/Offline Operation" on page 4-6).

The tabs on this dialog are used to display different pages that contain various parameters related to the given process. Refer to the section "Add Process and Edit Process Dialog Pages" on page 4-28 for complete descriptions.

The buttons along the bottom of the dialog control actions to be taken after you have entered the scheduling parameters for a specified program:

OK

Clicking on the OK button makes the specified changes, if any, and closes the dialog.

Reset

Clicking on the Reset button resets the fields to the original values without closing the dialog.

Cancel

Clicking on the Cancel button resets the fields to the original values if changed and closes the dialog.

Help

Clicking on the Help button displays help for the dialog.

Add Process and Edit Process Dialog Pages

The bottom portion of the Add Process and Edit Process dialogs is divided into a number of pages that contain various parameters related to the given process. These pages are:

- Runtime Properties (see "Runtime Properties" on page 4-29)
- FBS Schedule (see "FBS Schedule" on page 4-31)
- I/O and Debug (see "I/O and Debug" on page 4-33)

Runtime Properties

The Runtime Properties page of the Add Process and Edit Process dialogs (see "Add Process and Edit Process Dialog Pages" on page 4-28) allows the user to specify the working directory, CPU bias, scheduling policy and priority of the process.

The Runtime Properties page of the Edit Process dialog is shown in Figure 4-17:

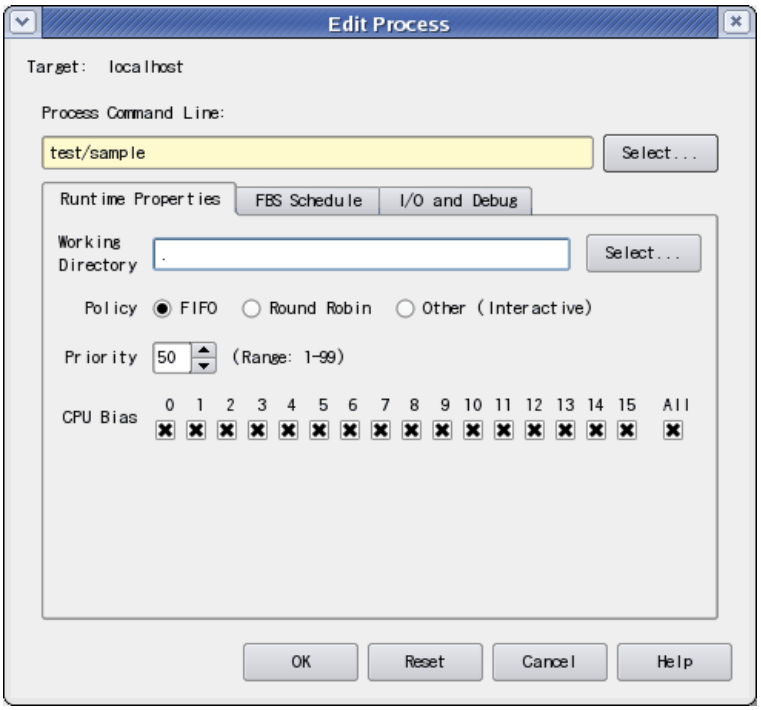


Figure 4-17. Add Process and Edit Process Dialog - Runtime Properties Page

Descriptions of the fields on the Runtime Properties page follow:

Working Directory

You may specify the path to the working directory in which the scheduled process will run. This path must be a valid directory name on the *target system*. If the target system is not available, or if the Scheduler is Offline (see "Online/Offline Operation" on page 4-6), the path must be a valid directory on the target system at the time the program scheduler is created (i.e. when the Set up button is pressed in the Scheduler Control Area).

Select...

You may use the Select button to bring up the Select a working directory dialog to specify the working directory for the program you want to configure under the *FBS*.

Policy

This panel of radio buttons allows you to select the POSIX scheduling policy for the specified program. The options are as follows: the FIFO (first–in–first–out) scheduling policy, the Round Robin scheduling policy, and the Other (interactive) scheduling policy.

Priority

The range of priority values that you can enter is governed by the scheduling policy specified. You can determine the allowable range of priorities associated with each policy (FIFO, Round-Robin, or Other) by invoking the **`run(1)`** command from the shell and not specifying any options or arguments (see the corresponding system manual page for an explanation of this command). Higher numerical values correspond to more favorable scheduling priorities.

NightSim displays the range of priority values that you can enter next to the Priority field.

NOTE

If the Priority of a process is changed while the scheduler is running, the Monitor Area will not reflect the updated value until the next Monitor Area update after the process runs again.

CPU Bias

This panel of check boxes allows you to select the processor or processors on which the program can be scheduled. Click on one or more boxes, where the label 0 represents the first logical CPU, 1 represents the second, and so on. To select all CPUs on your system, click on the All box. To deselect, click on the checked box.

NOTE

If the CPU Bias of a process is changed while the scheduler is running, the Monitor Area will not reflect the updated value until the next Monitor Area update after the process runs again.

NOTE

You cannot change the CPU bias of **`/idle`** while the scheduler is active because it is not a real process. You must either add a new process with the desired CPU bias, or make the change while the scheduler is inactive. See "How Is Idle Time Monitored?" on page 5-5 for more information.

FBS Schedule

The FBS Schedule page of the Add Process and Edit Process dialogs (see "Add Process and Edit Process Dialog Pages" on page 4-28) allows the user to specify parameters such as the starting cycle, period, and overrun limits for the given process.

The FBS Schedule page of the Edit Process dialog is shown in Figure 4-18:

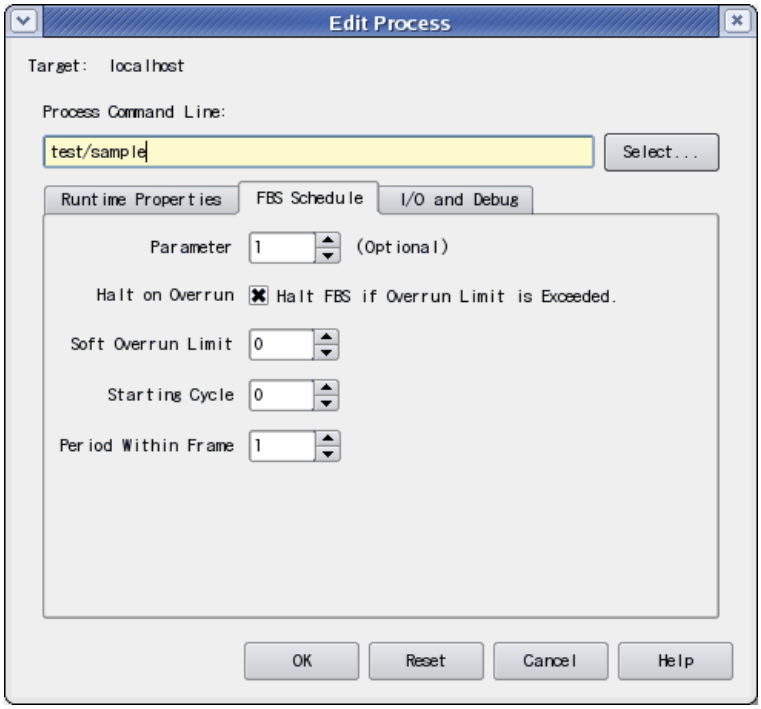


Figure 4-18. Add Process and Edit Process Dialogs - FBS Schedule Page

Descriptions of the fields on the FBS Schedule page follow:

Parameter

This field allows you to pass an integer value to a process that is scheduled on a frequency–based scheduler. The value must be a 32-bit decimal value. The default is no value. The process can retrieve this value by using the **`sched_fbsqry(3)`** library routine.

Halt on Overrun

This check button allows you to indicate whether or not the selected scheduler should be stopped in the event that the specified program causes a *frame overrun*. Check this field if you wish the scheduler to be stopped.

NOTE

This may useful in conjunction with scheduling a process under the NightView debugger, making it easier to stop a scheduled process at the source of an overrun and determine the activity which caused the overrun to occur.

Soft Overrun Limit

This field allows you to specify a consecutive overrun limit count that this process will tolerate and have processed as soft overruns by the kernel. *Soft overruns* are catastrophic failures only if the process reaches its limit on the number of soft overruns tolerated. This soft overrun limit defaults to 0 for each process.

Starting Cycle

This field allows you to specify the first *minor cycle* in which the specified program is to be wakened in each *major frame*. Enter a number ranging from zero to the total number of minor cycles per frame minus one. (The total number of minor cycles per frame is defined when you use the Scheduler Definition Area to configure a scheduler. See "Using the Scheduler Definition Area" on page 4-16 for details.)

NOTE

You must enter a starting cycle of zero for unscheduled processes (for an explanation of unscheduled processes, see "Using the Monitor Area" on page 5-1 and the section entitled "Monitoring Unscheduled Processes" on page 5-5).

Period Within Frame

This field allows you to establish the frequency with which the specified program is to be wakened in each *major frame*. A period of one indicates that the specified program is to be wakened every *minor cycle*; a period of two indicates that it is to be wakened once every two minor cycles, a period of three once every three minor cycles, and so on. Enter the number of minor cycles representing the frequency with which you wish the program to be wakened. This number can range from zero to the number of minor cycles that compose a frame on the scheduler. (The total number of minor cycles per frame is defined when you use the Scheduler Definition Area to configure a scheduler. See "Using the Scheduler Definition Area" on page 4-16 for details.)

NOTE

You must enter a period of zero for unscheduled processes (for an explanation of unscheduled processes, see the section entitled "Monitoring Unscheduled Processes" on page 5-5).

NOTE

If the Period Within Frame of a process is changed while the scheduler is running, the Monitor area will not reflect the updated value until the next Monitor area update after the process runs again.

I/O and Debug

The I/O and Debug page of the Add Process and Edit Process dialogs (see "Add Process and Edit Process Dialog Pages" on page 4-28) allows the user to specify where to read input from, where to send output, and if the process is to be scheduled within a NightView dialog or within an **`xterm(1)`** window.

The I/O and Debug page of the Edit Process dialog is shown in Figure 4-19:

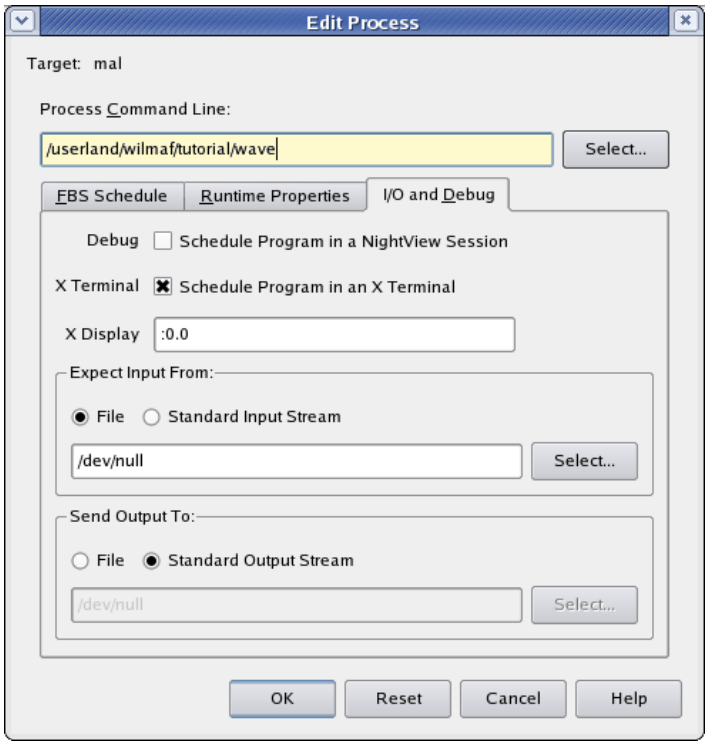


Figure 4-19. Add Process and Edit Process Dialog - I/O and Debug Page

Descriptions of the fields on the I/O and Debug page follow:

Debug

Schedule this process under the NightView debugger. When the Schedule program in a NightView session checkbox is checked, the process will be stopped at the point it begins execution, allowing early intervention for the purpose of setting

breakpoints, watchpoints, etc. A **resume** command to NightView is required to resume the execution of the scheduled process.

NOTE

Scheduling processes under the FBS requires certain capabilites (see "Capabilities" on page 2-2). It is the responsibility of the user to ensure that appropriate capabilities are granted to the NightView session.

X Terminal

Schedule the process in an **xterm(1)** window. An **xterm** window may be used in conjunction with scheduling the process under NightView as well, if so desired (see Debug above). This permits the scheduled process to interact with the terminal services provided by the **xterm**.

X Display

The X Display field may be used to provide a valid X display for use when scheduling the process in an Xterm window. When the NightSim host display name is unreachable from a remote target system, or when running NightSim with multiple displays, use this field to specify the name of the correct X display to use for **xterm(1)**.

Expect Input From:

Select the input source, if any, for the scheduled process. By default, as in previous versions of NightSim, input is taken from the file, **/dev/null**. When either the Debug or X Terminal checkboxes are selected, the default is to permit input to come from the Standard Input Stream, but you may choose File input in either case if your scheduled program needs a fixed input stream (see Input File below).

Input File

If File is selected for Expect Input From, this field specifies the name of the input file which the scheduled process will use. By default, the file is **/dev/null**. See "Pathname Expansion" on page 4-35.

If the NightSim Scheduler is Online (see "Online/Offline Operation" on page 4-6), the pathname given must be a valid file on the target system with sufficient read permissions.

If the NightSim Scheduler is Offline, the path must represent a valid readable file at the time the program scheduler is created (i.e. when the NightSim Scheduler transitions to an Online mode of operation).

Select...

You may use the Select button to bring up the Select an Input File dialog to specify the name of the input file which the scheduled process will use.

Send Output To:

Select the way to manage output from your scheduled process. If neither the Debug and X Terminal checkboxes are selected, you may select File output, where the default file is **/dev/null**. Output may be sent to an actual file specified in the Output File field, if desired. By default, output is sent to the Standard Output Stream of the scheduled process.

Output File

If File is selected for Send Output To, this field specifies the name of the output file to generate. The output file name may be a full pathname or may be a pathname relative to the Working Directory specified for the scheduled process. See "Pathname Expansion" on page 4-35.

Select...

You may use the Select button to bring up the Select an Output File dialog to specify the name of the output file that will be generated.

Pathname Expansion

When entering a process or file name in the Add Process or Edit Process dialog, the user can make use of the following conventions:

- if the pathname begins with **/**, it is assumed to be fully rooted
- if the pathname begins with **.**, **..**, or a simple file or directory name, it is assumed to be relative to the path in the Working Directory field
- if the pathname begins with **~**, it is interpreted according to the values of the following environment variables *on the NightSim host*:
  - **~**, **~/**, or **~***user* (where *user* is the username of the user running NightSim), the login directory of the user is substituted
  - **~***otheruser* (where *otheruser* is a user other than the user running NightSim), the login directory of *otheruser* is substituted
  - **~+**, the value of `$PWD` on the NightSim host is substituted.
  - ~-, the value if `$OLDPWD` on the NightSim host is substituted
- if any part of the pathname contains an environment variable reference of the form `$`*name*, the value of `$`*name* on the NightSim host is substituted

The pathnames will be expanded and substitutions made at the time the Apply or OK button is pressed.

5
Using the Monitor Area

This chapter describes the *performance monitor* and explains how to monitor *FBS*-scheduled processes, a processor's *idle time*, and unscheduled processes. It introduces the components of the Monitor area of the NightSim window and describes the use of each component.

Understanding the Performance Monitor

The performance monitor is a mechanism that enables you to monitor *FBS*–scheduled processes' utilization of a CPU.

The performance monitor provides you with the ability to:

- Obtain performance monitor values by process or processor
- Start and stop performance monitoring by process
- Clear performance monitor values by processor

You also have the ability to set the *timing mode* under which the performance monitor is to run. You can select one of two modes: one that includes time spent servicing interrupts in performance monitor timing values and one that excludes time spent servicing interrupts from those values.

When the performance monitor timing mode is set to include interrupt time, a process's user and system times will total the elapsed time that accrues when the process is the currently running process. This elapsed time includes time spent servicing interrupts and performing context switches. Time spent servicing interrupts is added to the process's system time. Time spent switching to a new process is included in the new process's system time.

When the performance monitor timing mode is set to exclude interrupt time, a process's user and system times will total the time that accrues when the process is the currently running process. This time excludes time spent servicing interrupts, but it includes time spent performing context switches. Time spent switching to a new process is included in the new process's system time.

What Values Are Monitored?

The *performance monitor* keeps track of the time that a process spends running from the time that it is wakened by a frequency–based scheduler until it calls **`fbswait`**, excluding time during which the scheduler was stopped. Time is calculated from the frame and cycle count and the clock period at the time the sample performance monitor data is taken, measured in microseconds. One instance of a process's being wakened by a scheduler is referred to as an iteration or a cycle. Performance monitor values for *FBS*–scheduled processes are reported both in terms of cycles, or *iterations*, and in terms of *major frames*. They reflect what has happened since the last time that performance monitor values were cleared and performance monitoring was enabled.

When performance monitoring is enabled for an *FBS*–scheduled process, NightSim maintains the following values for each process:

Process Parameters

Target System

: The system on which this process is running

PID

: The process's process identification number

FPID

: The process's unique *frequency-based scheduler* identifier

Program Name

: The process's name

Policy

: The process's scheduling policy

Priority

: The process's scheduling priority

Start Cycle

: The number of the first *minor cycle* in which the process is to be wakened in each frame

Period

: The frequency with which the process is to be wakened in each *major frame*

CPU Bias

: The process's CPU mask

General Statistics

Parameter

An interval value passed to the process when it begins execution

Total iterations, cycles

The number of times that the process has been wakened by the scheduler

Last time

The amount of time that the process has spent running from the last time that it has been wakened by the scheduler until it has called **`fbswait`**

Total time

The total amount of time that the process has spent running in all cycles

Average time

The average amount of time that the process has spent running in all cycles, or *iterations*. This value is obtained by dividing the value reported in the Total Time column by the value reported in the Iterations column.

Soft Overruns

The number of times that the process has caused consecutive "soft" frame overruns.

Overruns

The number of times that the process has caused consecutive "hard" frame overruns.

Overrun Limit

The number of soft overruns allowed before halting the scheduler.

Halt on Overruns?

Whether to halt the scheduler when the soft overrun limit is reached.

Percent of Period Used

This value is maintained only if the interrupt source for the *frequency-based scheduler* on which the process is scheduled is a real-time clock. It is obtained by dividing the value reported in the Average Time column by the amount of time that is available to the process. The amount of time available to the process is calculated from its *period* and the length of a clock cycle.

Interrupt Time Inclusion

Indicates whether time spent processing interrupts is included in performance monitor timing values.

Minimum Values

Minimum cycle time

The least amount of time that the process has spent running in a cycle

Minimum cycle frame

The number of the *major frame* in which the *minimum cycle time* has occurred

Minimum cycle cycle

The number of the *minor cycle* in which the *minimum cycle time* has occurred

Minimum frame time

The least amount of time that the process has spent running during a *major frame*

Minimum frame frame

The number of the *major frame* in which the *minimum frame time* has occurred

Maximum Values

Maximum cycle time

The greatest amount of time that the process has spent running in a cycle

Maximum cycle frame

The number of the *major frame* in which the *maximum cycle time* has occurred

Maximum cycle cycle

The number of the *minor cycle* in which the *maximum cycle time* has occurred

Maximum frame time

The greatest amount of time that the process has spent running during a *major frame*

Maximum frame frame

The number of the *major frame* in which the *maximum frame time* has occurred

Monitoring Idle Time

The *performance monitor* provides you with the capability of monitoring a processor's idle time. Idle time refers to the time that the CPU is not busy. By monitoring a proces-

sor's idle time, you can determine the amount of CPU time that is available to be allocated to additional processes.

Procedures for monitoring idle time are described in "How Is Idle Time Monitored?" on page 5-5.

How Is Idle Time Monitored?

You can monitor a particular processor's *idle time* if you add the process **`/idle`** to a frequency–based scheduler and schedule it on the desired processor. You can also monitor idle time for a number of different processors by adding **`/idle`** to a selected *frequency–based scheduler* with multiple CPUs selected. One **`/idle`** process will be added per CPU selected.

NOTE

You can schedule **`/idle`** on a particular processor for a particular target only once. This is a system-wide restriction. Only one user per processor per target may schedule **`/idle`**.

To add **`/idle`** to a frequency–based scheduler, use the Processes Area of the NightSim window (see "Using the Processes Area" on page 4-20 for details). For an explanation of the procedures for adding a program to a scheduler, refer to "Add Process" on page 4-24

When you use NightSim to add **`/idle`** to a frequency–based scheduler, the only parameter that you may specify is the CPU bias mask. The CPU mask for **`/idle`** may specify one or more CPUs. If multiple CPUs are selected, **`/idle`** will be scheduled on each selected CPU. The default scheduling priority is zero. The starting base cycle is zero, and the period is one. **`/idle`** will be scheduled every *minor cycle*, starting with the first minor cycle in each *major frame*.

You can view scheduling information for **`/idle`** in the same way that you view it for other *FBS*–scheduled processes—in the Processes Area of the NightSim window (see "Using the Processes Area" on page 4-20 for details).

Monitoring Unscheduled Processes

The *performance monitor* provides you with the additional capability of monitoring the performance of unscheduled processes. Unscheduled processes are those that are not wakened by the scheduler and do not call **`fbswait`**; they are not scheduled to run at a certain frequency. To be able to obtain *performance monitor* values for such processes, you must first add them to a *frequency–based scheduler* and specify a starting base cycle of zero and a *period* of zero. The other scheduling parameters that you must specify include the process's scheduling priority and the CPU on which it is to be scheduled. You can optionally specify a decimal value to be passed to a process that is scheduled on a frequency–based scheduler. The Halt on overrun flag does not apply to an unscheduled process.

You can add unscheduled processes to a frequency–based scheduler by using the Processes Area (see "Using the Processes Area" on page 4-20 for details). For an explanation of the procedures for adding a program to a scheduler, refer to "Add Process" on page 4-24.

You can view scheduling information for unscheduled processes in the same way that you view it for other *FBS*–scheduled processes—in the Processes Area (see "Using the Processes Area" on page 4-20 for details).

The *performance monitor* values that are maintained for unscheduled processes include the following:

- Total time

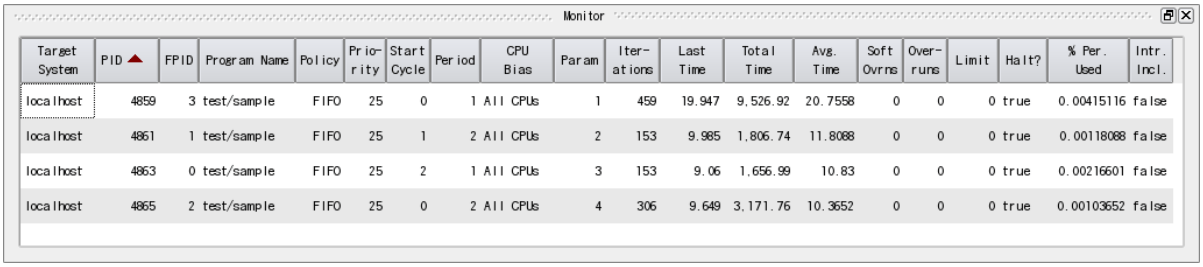
You can obtain this value if you enable performance monitoring for the processes. Procedures for enabling performance monitoring and retrieving *performance monitor* values are described in detail in "The Monitor Menu" on page 5-8.

Introduction to the Monitor Area

The Monitor area allows you to start and stop performance monitoring on *FBS*-scheduled processes scheduled on one or more processors, clear *performance monitor* values for all FBS-scheduled processes scheduled on one or more processors, and display *performance monitor* values for selected FBS-scheduled processes.

The Monitor area appears by clicking on the Monitor tab at the bottom of the screen or by selecting Monitor from the View menu.

Figure 5-1 is an example of the Monitor area.



Target System	PID	FPID	Program Name	Policy	Priority	Start Cycle	Period	CPU Bias	Param	Iterations	Last Time	Total Time	Avg. Time	Soft Ovrns	Over-runs	Limit	Halt?	% Per. Used	Intr. Incl.
localhost	4859	3	test/sample	FIFO	25	0	1	All CPUs	1	459	19.947	9,526.92	20.7558	0	0	0	true	0.00415116	false
localhost	4861	1	test/sample	FIFO	25	1	2	All CPUs	2	153	9.985	1,806.74	11.8088	0	0	0	true	0.00118088	false
localhost	4863	0	test/sample	FIFO	25	2	1	All CPUs	3	153	9.06	1,656.99	10.83	0	0	0	true	0.00216601	false
localhost	4865	2	test/sample	FIFO	25	0	2	All CPUs	4	306	9.649	3,171.76	10.3652	0	0	0	true	0.00103652	false

Figure 5-1. Monitor Area

The Monitor area also allows you to specify whether performance monitoring information is to be displayed in the Monitor area or written to a selected file. In either case, you may specify the types of information that are to be reported and the resolution with which performance monitor timing values are displayed (for example, microseconds, tens of microseconds, milliseconds).

NOTE

If the Priority, Period, or CPU Bias mask of a process are changed while the scheduler is running, the Monitor area will not reflect the updated values until the next Monitor area update after the process runs again.

It is important to note that the Monitor area can be “torn off” to reside in its own window separate from NightSim by clicking on the dashed line at the top of the area and dragging to another location on the desktop.

The Monitor menu contains general commands related to performance monitoring.

Columns in the Monitor area can be resized or hidden using selections from the context menu. The context menu is displayed by right-clicking in the Monitor area or on a specific column heading.

The Monitor Menu

The Monitor menu provides options for disabling monitoring, setting the timing mode, changing the display resolution, selecting values, saving output to a file, and clearing data.

The Monitor menu appears on the NightSim window menu bar (see "The Menu Bar" on page 3-2). Figure 5-2 presents this menu.

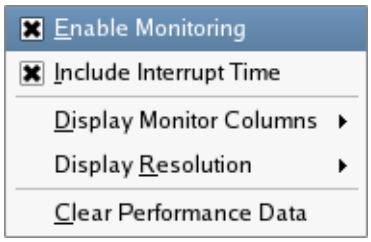


Figure 5-2. Monitor Menu

Descriptions of the options on the Monitor menu follow.

Enable Monitoring

Mnemonic: E

This item indicates the state of performance monitoring. By default, monitoring is enabled for all FBS-scheduled processes and this box is checked. To disable Performance Monitoring for all processes, uncheck this box.

Include Interrupt Time

Mnemonic: I

This item allows you set the *timing mode* under which the performance monitor is to run, indicating that time spent processing interrupts is to be **included** in *performance monitor* timing values. This is a system-wide setting that affects all processes on the *target systems* that have processes attached to the specified scheduler.

Exclude Interrupt Time

This item allows you to set the *timing mode* under which the performance monitor is to run, indicating that time spent processing interrupts is to be **excluded** from *performance monitor* timing values. This is a system-wide setting that affects all processes on the *target systems* that have processes attached to the specified scheduler.

Display Monitor Columns

Mnemonic: D

This option allows you to specify the types of *performance monitor* values that are to be displayed in the Monitor area. When you select this option, NightSim displays the menu that Figure 5-3 presents.

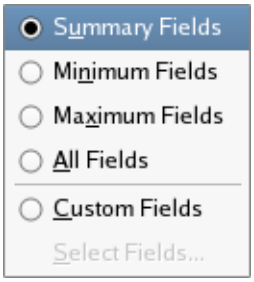


Figure 5-3. Display Monitor Columns Menu

Descriptions of the options contained in the Display Monitor Columns menu follow.

Summary Fields

This option specifies the following types of values: *target system*, process parameters (*FPID*, program name, and CPU mask) and general statistics. Refer to "What Values Are Monitored?" on page 5-2 for a complete description of all types of *performance monitor* values.

Minimum Fields

This option specifies the following types of values: *target system*, process parameters (*FPID*, program name, and CPU mask), general statistics (*iterations*) and minimum values. Refer to "What Values Are Monitored?" on page 5-2 for a complete description of all types of *performance monitor* values.

Maximum Fields

This option specifies the following types of values: *target system*, process parameters (*FPID*, program name, and CPU mask), general statistics (*iterations*) and maximum values. Refer to "What Values Are Monitored?" on page 5-2 for a complete description of all types of *performance monitor* values.

All Fields

This option specifies all types of values: *target system*, process scheduling parameters, general statistics, minimum values, and maximum values. Refer to "What Values Are Monitored?" on page 5-2 for a complete description of all types of *performance monitor* values.

Custom Fields

This option allows you to select any of the types of values that the *performance monitor* facility maintains.

When you select this option initially, NightSim displays the dialog that Figure 5-4 presents.

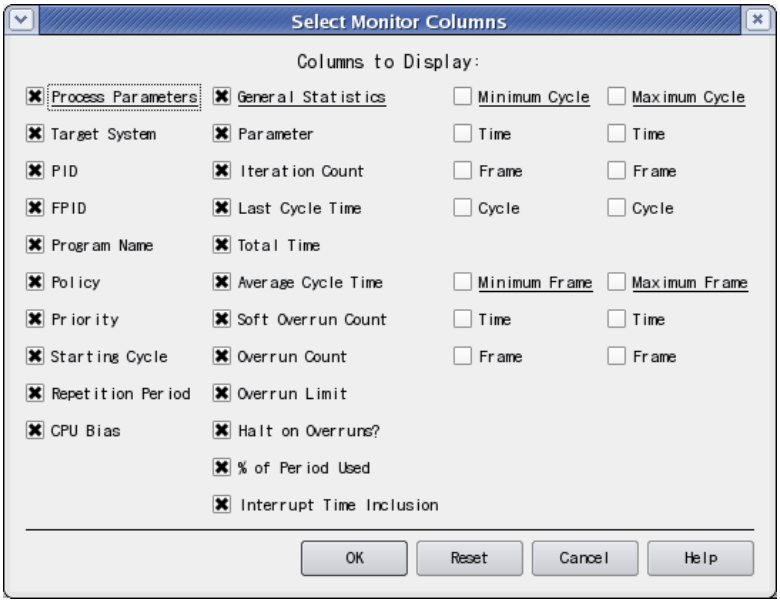


Figure 5-4. Monitor Columns to Display Dialog

You use the panels of check buttons to select the types of *performance monitor* values that you wish to save. Click on one or more buttons as desired. For a description of each value, see the section "What Values Are Monitored?" on page 5-2.

After you have selected the types of values that you wish to save, select OK to apply the selections and close the dialog box, or you can reset the original settings, cancel the selections, or display help related to the dialog.

Select Fields...

When Custom Fields is enabled and you wish to change the selection of fields, this option displays the Columns to Display dialog so that changes can be made.

Display Resolution

Mnemonic: R

This option allows you to control the precision with which NightSim reports the *performance monitor* timing values. When you select this option, NightSim displays the menu that Figure 5-5 presents.

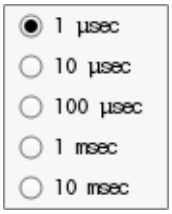


Figure 5-5. Resolution Options

You may specify the precision in microseconds, tens of microseconds, hundreds of microseconds, milliseconds, or tens of milliseconds by clicking on the radio button.

NOTE

As the values of times increase in a long-running simulation, the resolution option value is automatically increased by NightSim in order to avoid overflowing the display.

Clear Performance Data

Mnemonic: C

This option clears all *performance monitor* values for all *FBS*-scheduled processes on all processors.

A
NightStar Licensing

NightStar RT uses the NightStar License Manager (NSLM) to control access to the NightStar RT tools.

License installation requires a licence key provided by Concurrent (see “License Keys” on page A-1).The NightStar RT tools request a licence (see “License Requests” on page A-2) from a license server (see “License Server” on page A-2).

Two license modes are available, fixed and floating, depending on which product option you purchased. Fixed licenses can only be served to NightStar RT users from the local system. Floating licenses may be served to any NightStar RT user on any system on a network.

Tools are licensed per system, per concurrent user. A single license is shared among any or all of the NightStar RT tools for a particular user on a particular system. The intent is to allow *n* developers to fully utilize all the tools at the same time while only requiring *n* licenses. When operating the tools in remote mode, where a tool is launched on a local system but is interacting with a remote system, licenses are required only from the host system.

You can obtain a license report which lists all licenses installed on the local system, current usage, and expiration date for demo licenses (see “License Reports” on page A-3).

The default configuration includes a strict firewall which interferes with floating licenses. See “Firewall Configuration for Floating Licenses” on page A-3 for information on handling such configurations.

See “License Support” on page A-4 for information on contacting Concurrent for additional assistance with licensing issues.

License Keys

Licenses are granted to specific systems to be served to either local or remote clients, depending on the license model, fixed or floating.

License installation requires a license key provided by Concurrent. To obtain a license key, you must provide your system identification code. The system identification code is generated by the **`nslm_admin`** utility:

```
nslm_admin --code
```

System identification codes are dependent on system configurations. Reinstalling Linux on a system or replacing network devices may require you to obtain new license keys.

To obtain a license key, use the following URL:

http://www.ccur.com/NightStarRTKeys

Provide the requested information, including the system identification code. Your license key will be immediately emailed to you.

Install the license key using the following command:

```
nslm_admin --install=xxxx-xxxx-xxxx-xxxx-xxxx
```

where *xxxx-xxxx-xxxx-xxxx-xxxx* is the key included in the license acknowledgment email.

License Requests

By default, the NightStar RT tools request a license from the local system. If no licenses are available, they broadcast a license request on the local subnet associated with the system's hostname.

You can control the license requests for an entire system using the **/etc/nslm.config** configuration file.

By default, the **/etc/nslm.config** file contains a line similar to the following:

```
:server @default
```

The argument @default may be changed to a colon-separated list of system names, system IP addresses, or broadcast IP addresses. Licenses will be requested from each of the entities found in the list, until a license is granted or all entries in the list are exhausted.

For example, the following setting prevents broadcast requests for licenses, by only specifying the local system:

```
:server localhost
```

The following setting requests a license from **server1**, then **server2**, and then a broadcast request if those fail to serve a license:

```
:server server1:server2:192.168.1.0
```

Similarly, you can control the license requests for individual invocations of the tools using the **NSLM_SERVER** environment variable. If set, it must contain a colon-separated list of system names, system IP addresses, or broadcast IP addresses as described above. Use of the **NSLM_SERVER** environment variable takes precedence over settings defined in **/etc/nslm.config**.

License Server

The NSLM license server is automatically installed and configured to run when you install NightStar RT.

The **nslm** service is automatically activated for run levels 2, 3, 4, and 5. You can check on these settings by issuing the following command:

```
/sbin/chkconfig --list nslm
```

In rare instances, you may need to restart the license server via the following command:

```
/sbin/service nslm restart
```

See **nslm(1)** for more information.

License Reports

A license report can be obtained using the **nslm_admin** utility.

```
nslm_admin --list
```

lists all licenses installed on the local system, current usage, and expiration date (for demo licenses). Use of the **--verbose** option also lists individual clients to which licenses are currently granted.

Adding the **--broadcast** option will list this information for all servers that respond to a broadcast request on the local subnet associated with the system's hostname.

See **nslm_admin(1)** for more options and information.

Firewall Configuration for Floating Licenses

RedHawk does not support a firewall configuration by default, because iptables support is disabled. However, it is possible to build a custom kernel with iptables support enabled. If that is done, and floating licenses are used, the iptables firewall rules must be configured to allow the license requests and responses to pass.

If the system with iptables support and firewall rules is serving licenses, then the firewall rules must be arranged to allow license requests on UDP port 25517 and TCP port 25517 from any systems that will make license requests. For example, in a simple firewall, rules like the following, inserted before any DROP or REJECT rules, might work:

```
iptables -A INPUT -p udp -m udp -s subnet/mask --dport 25517 -j ACCEPT
iptables -A INPUT -p tcp -m tcp -s subnet/mask --dport 25517 -j ACCEPT
```

If the system with iptables support and firewall rules is running NightStar RT tools and receiving floating licenses, then the firewall rules must be arranged to allow license responses on UDP port 25517 from any system serving licenses. For example, in a simple firewall, rules like the following, inserted before any DROP or REJECT rules, might work:

```
iptables -A INPUT -p udp -m udp -s subnet/mask --sport 25517 -j ACCEPT
```

License Support

For additional aid with licensing issues, contact the Concurrent Software Support Center at our toll free number 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822. The Software Support Center operates Monday through Friday from 8 a.m. to 5 p.m., Eastern Standard Time.

You may also submit a request for assistance at any time by using the Concurrent Computer Corporation web site at **http://www.ccur.com/isd_support_contact.asp** or by sending an email to **support@ccur.com**.

B
Kernel Dependencies

Concurrent's RedHawk kernel provides features and performance gains that are critical for the optimal operation of the NightStar RT tools.

The NightStar RT tools can operate in a host-only mode on Red Hat systems without Concurrent's RedHawk kernel, cross-targeting to RedHawk systems.

Additionally, the NightStar RT tools can function on Red Hat systems without the RedHawk kernel, but will lack the numerous advantages afforded by running with it.

The following sections describe the additional functionality and capabilities of the NightStar RT tools when running Concurrent's RedHawk kernel

Advantages for NightView

The following advantages are afforded NightView when Concurrent's RedHawk kernel is running:

- Application speed conditions

  Provides "execution-speed" patches, conditions, and ignore counts.

- Signal handling

  Allows NightView to pass signals directly to a particular process, avoiding context switching.

Advantages for NightTrace

The following advantage is afforded NightTrace when Concurrent's RedHawk tracing kernel is running:

- Kernel tracing

  Users of NightTrace gain the ability to obtain kernel trace data and combine that with user trace data. Kernel tracing is an incredibly powerful feature that not only provides insight into the operating system kernel but also provides useful information relating to the execution of user applications.

  The RedHawk kernel is provided in three flavors:

  - Tracing

- Debug
- Plain

The Tracing and Debug flavors provide the features required for NightTrace kernel tracing. These kernels can be selected at boot-time from the boot-loader menu.

Advantages for NightProbe

The following advantages are afforded NightProbe when Concurrent's RedHawk a RedHawk or SLERT kernel is running:

- Minimal intrusion

  Allows NightProbe to read and write variables without stopping the process for each sample or write operation.

- Sampling performance

  Allows NightProbe to use direct memory fetches for data sampling (as opposed to programmed I/O) which is important for high-rate data acquisition.

- Concurrent debugging/probing

  Allows NightProbe to probe programs already under the control of a debugger or another NightProbe session.

- PCI Device probing

  Allows NightProbe to probe PCI device memory via the Base Address Register (BAR) file system.

Advantages for NightTune

The following advantage is afforded NightTune when Concurrent's RedHawk a RedHawk or SLERT kernel is running:

- Context switch rate

  Allows NightTune user to display the context switch counts per CPU instead of for the overall system.

- CPU shielding

  Individual CPUs can be shielded from interrupts and processes allowing CPUs to be dedicated solely to specific interrupts and processes that are bound to the CPU.

- CPU sibling interference

  Individual CPUs can be marked down to avoid interfering with hyperthreaded sibling CPUs and dual-core sibling CPUs. Hyperthreaded CPUs share all the resources of their sibling CPU. Dual-core CPUs share the CPU cache and a path to memory with their sibling CPU.

- Detailed memory information

  Detailed process memory descriptions include the residency and lock state of any page in a process, and their association with physical memory pools for NUMA systems.

Frequency Based Scheduler

The Frequency Based Scheduler is only available on RedHawk systems from Concurrent Computer Corporation. It is required for all NightSim usage.

NightSim is only included in NightStar distributions intended for use on RedHawk systems.

PCI Bar File System

The PCI Bar File System is only available with the RedHawk kernel from Concurrent Computer Corporation and SLERT versions 1.0-1.6 kernel from Novell.

On other systems, PCI Device probing will be disabled within NightProbe.

C
Configuration Files

NightSim allows the user to save the configuration to a file with the following format:

```
Scheduler <key> {
   cycles <n>;                  // cycles per frame
   tasks total <n>;             // total number of tasks allowed in scheduler
   tasks per cycle <n>;         // optional, defaults to total tasks
   permissions owner=<perms>;   // permissions, see below

   timing source {
      host "<name>";            // timing-host name
      EOC ;                     // if end-of-cycle timing
      AvailableRTC;             // if First Available RTC Timing
      clock "<name>";           // if rtc, simple name of clock
      eti "<name>";             // if eti, simple name of eti
      other "<name>";           // full pathname of device
      clock duration <n>;       // if rtc, clock duration in microseconds
      clock count <n>;          // if rtc, ticks per minor cycle.
   }

   process "<program_name>" on "<target_name>" {
      Arguments {               // If arguments are present
         "arg 1"
         ["arg 2"]
         ...
      }
      NightView <boolean> ;      // schedule under NightView, {True | False}
      xterm <boolean> ;          // run under an xterm(1), {True | False}
      dir "<name>" ;              // full pathname of working directory on target
      input <style> ;           // input source, {File | Standard}
      infile "<name>" ;         // full pathname of input file
      output <style> ;          // output destination, {File | Standard}
      outfile "<name>" ;        // full pathname of output file
      privilege "<privlist>" ;  // working privileges, see below
      policy <p>;               // scheduling policy, p = {R | O | P}
      priority <n>;             // scheduling priority
      parameter <n>;            // program parameter.  if absent, -1
      soft overruns <n>;        // soft overrun limit.  if absent, no limit
      cpu <mask>;               // cpu bias mask, see below
      start cycle <n>;          // starting cycle
      period <n>;               // period
      [cwd "<path>";]           // initial working directory
      [display "<display_name>";] // X Display Name
   } ...

}
```

NOTE

The values for <name>, <program_name>, <target_name>, and <privlist> must be enclosed in double quotes (").

The <perms> specification is a sequence of one or more of the following:

```
owner=<permspec>
group=<permspec>
other=<permspec>
```

where <permspec> is either 'r', 'w', or 'rw'.

The <boolean> specifications should have a simple boolean keyword:

`True`	the feature is enabled
`False`	the feature is disabled

The <style> specifications should have one of the following values:

`File`	input/output should be redirected from/to a file on the target system. A name must be provided.
`Standard`	I/O should be performed to/from standard I/O streams, Standard Input, or Standard Error.

NOTE

If both `xterm` and `NightView` boolean parameters are `False`, then the <style> specifications for `input` and `output` must both be `File`.

The <privlist> specification should contain a comma-separated list of simple privilege names. For example:

```
"owner,dev,rtime"
```

See **`intro(2)`** for more information on privileges.

The <mask> specification can be a comma-separated list of mask specifiers. Each specifier can be a CPU number (specified as an integer), a range of CPU numbers in the form <n>-<m>, or the keyword `active`, indicating all active CPUs on the system.

Configuration File Example

The following is an example of a configuration file.

```
Scheduler 2778 {
   cycles 4;
   tasks total 4;
   tasks per cycle 4;
   permissions owner=rw;

   timing source {
      host "localhost";
      clock "/dev/rcim/rtc2";
      clock duration 10;
      clock count 10;
   }

   process "./wave"
         on "localhost" {
      dir "." ;
      input File ;
      infile "/dev/null" ;
      output File ;
      outfile "/dev/null" ;
      policy F;
      priority 1;
      soft overruns 0;
      cpu 0;
      start cycle 0;
      period 1;
   }

   process "/idle"
         on "localhost" {
      dir "/" ;
      input File ;
      infile "/dev/null" ;
      output File ;
      outfile "/dev/null" ;
      policy F;
      priority 99;
      parameter 0;
      soft overruns 0;
      cpu 0;
      start cycle 0;
      period 1;
   }

}
```

Glossary

active scheduler window

A Scheduler window that is associated with a scheduler that exists on the system.

CPU bias

The processor or processors on which a program is scheduled to run.

end–of–cycle scheduling

A form of frequency–based scheduling in which scheduling is triggered when the last process that is scheduled to execute in the current *minor cycle* of the current *major frame* completes its processing.

FBS

Frequency-based scheduler.

FPID

A unique *frequency–based scheduler* process identifier that is returned when a program is scheduled on a frequency–based scheduler.

frame overrun

The condition that occurs when an *FBS*–scheduled process does not finish its processing before it is scheduled to run again.

frequency-based scheduler

A high resolution task synchronization mechanism that enables processes to run at user–specified frequencies.

host system

A system designated by a host name. A host may be a stand-alone computer system or a node in a network of computer system connected by ethernet. See *NightSim host*, *interrupt host*, *target system*.

idle time

Time during which the CPU is not busy.

inactive scheduler window

A Scheduler window that is not associated with an existing scheduler.

Interrupt host

The system on which the interrupt device physically resides.

iteration

One instance of a process's being wakened by a *frequency–based scheduler.*

last time

A value returned by the *performance monitor* indicating the amount of time that an *FBS*–scheduled process has spent running from the last time that it has been wakened by the scheduler until it has called **`fbswait`**.

major frame

One pass through all of the *minor cycles* with which a *frequency–based scheduler* is configured. A major frame has associated with it a duration, which is obtained by multiplying the duration of a minor cycle by the number of minor cycles per major frame.

maximum cycle cycle

A value returned by the *performance monitor* indicating the number of the *minor cycle* in which the *maximum cycle time* has occurred.

maximum cycle frame

A value returned by the *performance monitor* indicating the number of the *major frame* in which the *maximum cycle time* has occurred.

maximum cycle time

A value returned by the *performance monitor* indicating the greatest amount of time that an *FBS*–scheduled process has spent running in a cycle.

maximum frame frame

A value returned by the *performance monitor* indicating the number of the *major frame* in which the *maximum frame time* has occurred.

maximum frame time

A value returned by the *performance monitor* indicating the greatest amount of time that an *FBS*–scheduled process has spent running during a *major frame.*

minimum cycle cycle

A value returned by the *performance monitor* indicating the number of the *minor cycle* in which the *minimum cycle time* has occurred.

minimum cycle frame

A value returned by the *performance monitor* indicating the number of the *major frame* in which the *minimum cycle time* has occurred.

minimum cycle time

A value returned by the *performance monitor* indicating the least amount of time that an *FBS*–scheduled process has spent running in a cycle.

minimum frame frame

A value returned by the *performance monitor* indicating the number of the *major frame* in which the *minimum frame time* has occurred.

minimum frame time

A value returned by the *performance monitor* indicating the least amount of time that an *FBS*–scheduled process has spent running during a *major frame*.

minor cycle

The smallest unit of frequency maintained by a *frequency–based scheduler*. A minor cycle has associated with it a duration, which is the time that elapses between interrupts generated by the interrupt source that is attached to the scheduler. If the interrupt source is a real–time clock, the minor cycle duration is defined by specifying the number of clock counts per minor cycle and the number of microseconds per clock count.

NightSim host

The system from which NightSim was invoked.

NightSim server

The NightSim server is a program that executes on a *target system* and performs system-level actions on behalf of NightSim. Whenever NightSim needs to perform an operation on a target system, such as creating an *FBS*, scheduling a program on an FBS, or starting or stopping an FBS, NightSim makes a request of the NightSim server on that target system. NightSim automatically takes care of starting up a NightSim server on a target system when needed.

number of overruns

A value returned by the *performance monitor* indicating the number of times that an *FBS*–scheduled process has caused a soft or hard *frame overrun*.

performance monitor

A mechanism that makes it possible to monitor use of the CPU by processes that are scheduled on a *frequency–based scheduler*.

period

A *frequency–based scheduler* scheduling parameter that specifies the frequency with which a specified program is to be wakened in each *major frame*. A period of one indicates that the program is to be wakened every *minor cycle*; a period of two indicates that it is to be wakened once every two minor cycles; and so on.

privilege

A mechanism through which processes are allowed to perform sensitive operations or override system restrictions.

process dispatch latency

The time that elapses from the occurrence of an external event, which is signified by an interrupt, until the process that is waiting for that external event executes its first instruction in user mode.

process name

The name of the program.

RCIM

Real-Time Clocks and Interrupt Module - an optional hardware module which provides both real-time clocks and edge-triggered interrupt pins for the *host* to which it is attached, allowing those devices to be used as timing and interrupt sources for that host. RCIMs may be connected, and therefore synchronized, via RCIM cables. RCIM real-time clocks and edge-triggered interrupts may be used locally or as distributed timing devices.

scheduler key

A user–supplied numeric identifier for a *frequency–based scheduler.*

server

See *NightSim server.*

soft overrun

Soft overruns are a type of *frame overrun* which give the scheduled process a chance to recover and return to synchronization. When scheduling a process, you can specify a consecutive soft overrun limit count that this process will tolerate and have processed as soft overruns by the kernel. Soft overruns are catastrophic failures only if the process reaches its limit on the number of soft overruns tolerated. This soft overrun limit defaults to 0 for each process.

starting base cycle

A *frequency–based scheduler* scheduling parameter that specifies the first *minor cycle* in which an *FBS*–scheduled process is to be wakened in each *major frame.*

target system

Any system on which processes will be scheduled.

timing mode

The mode under which the *performance monitor* runs. It specifies whether time spent servicing interrupts is to be included in or excluded from performance monitor timing values.

total iterations

A value returned by the *performance monitor* indicating the number of times that an *FBS*–scheduled process has been wakened by the scheduler.

total time

A value returned by the *performance monitor* indicating the total amount of time that an *FBS*–scheduled process has spent running in all cycles.

unscheduled process

A process that is not wakened by the *frequency–based scheduler* and does not call **fbswait**; it is not scheduled to run at a certain frequency.

Index