NightView

Version 5.8 Release Notes (PowerMAX)

April 2004

0890395-5.8

READ ME BEFORE
INSTALLING
THIS PRODUCT

CONCURRENT
COMPUTER
CORPORATION

Copyright

Disclaimer

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

License

Trademark Acknowledgments

Contents

1.0. Introduction

NightView^{TM} is a general-purpose, graphical source-level debugging and monitoring tool specifically designed for real-time applications written in Ada, C, C++, and Fortran. NightView can monitor, debug, and patch multiple real-time processes running on multiple processors on the local system or on different targets with minimal intrusion. In addition to standard debugging capabilities, NightView supports application-speed eventpoint conditions, hot patches, synchronized data monitoring, exception handling and loadable modules.

2.0. Documentation

Table 2-1 lists the NightView 5.8 documentation available from Concurrent.

Table 2-1. NightView Version 5.8 Documentation

Manual Name	Pub. Number
NightView User's Guide	0890395-270
NightView Version 5.8 Release Notes (PowerMAX)	0890395-5.8
NightView Pocket Reference	0890475-070

Copies of the Concurrent documentation can be ordered by contacting the Concurrent Software Support Center. The toll-free number for calls within the continental United States is 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822 or 1-305-931-2408.

Additionally, the *NightView User's Guide* and *NightView Version 5.8 Release Notes (PowerMAX)* are available:

- online using the X Window System utility, **`nhelp`**
- on the Concurrent Computer Corporation web site at www.ccur.com

3.0. Prerequisites

Prerequisites for NightView Version 5.8 are as follows:

3.1. Software

- PowerMAX OS Version 4.3 or later

  To use watchpoints, you must have PowerMAX OS 4.3 patch-level 6 (base-006) or PowerMAX OS 5.1 or later. NightView disables the use of watchpoints if it doesn't detect the corresponding support in the kernel.

- X Window System (X11 Version 6.3 or later)
- An X™ server is required if you are using NightView's graphical user interface.
- Élan License Manager™ 5.0.1 or later (See *System Installation* on page 4 for details)

For more information about configuring your system to support NightView, see Appendix A, "System Resource Requirements" in the *NightView User's Guide* (0890395).

3.2. Hardware

- Computer Systems:

  Power Hawk 620 and 640

  Power Hawk 710, 720, 740 and 900

  PowerStack™ II and III

  Night Hawk® Series 6000

  TurboHawk™

  PowerMAXION™

- Board-Level Products:

  Motorola MVME2604

  Motorola MVME4604

4.0. System Installation

The NightView product is installed as standard PowerMAX OS software packages and utilizes the standard PowerMAX OS product installation mechanism, **pkgadd** (see **pkgadd(1)**).

The package names are **NightView** and **Nviewp**. These names are case-sensitive.

Please refer to the "Installing Add-on Software" chapter in the *System Administration Volume I* (0890429) manual and the *PowerMAX OS Release Notes* for instructions on software installation.

NightView may be installed in either the root directory or elsewhere. When you run **pkgadd(1M)** to install NightView, you are prompted to enter the name of the directory for installation. If you want to install in the root directory, just press the <return> key at the prompt. Otherwise, enter the name of the directory where you want NightView installed. If this directory does not exist, the installation procedures attempt to create it for you.

NOTE

> NightView consists of multiple packages. The **NightView** package must be installed on systems on which you want to run the NightView user interface. The **Nviewp** package must be installed on systems on which the user programs you want to debug will be running. Most customers will want both packages on the same system.

If the NightView package is installed before the **Nviewp** package, you will get a warning about the **Nviewp** package not being installed.

NightView **must** be run with the Élan License Manager. Follow the steps in the "Obtaining Licenses" section of the *Élan License Manager Release Notes* (0891055); the *feature alias* is **NightView**. If you are not already running the Élan License Manager, if you do not have a copy of the *Élan License Manager Release Notes*, or if you need a license key, contact Concurrent Software Distribution at 1-800-666-5405 (or 1-954-283-1836 outside the continental United States).

If your site has multiple license servers, and you need to indicate a server on a particular system, you can set the environment variable `POWERWORKS_ELMHOST` to the name of the server's system before invoking NightView. For more information, see the *Élan License Manager Release Notes* (0891055).

5.0. Overview of NightView 5.8

5.1. Enhancements

5.1.1. Save and Restore Layout

New menu items let you save and restore the layout of data items in a Data Window. See the section titled “Debug Display Menu” in Chapter 9 of the *NightView User’s Guide* (0890395).

5.1.2. Eventpoint Warning

NightView gives a caution if an eventpoint is being set on an address that does not appear to be an instruction boundary. See the section titled “Eventpoints” in Chapter 9 of the *NightView User’s Guide* (0890395).

5.1.3. with-translations Parameter

The `with-translations` parameter to the **`core-file`** command can be used to debug a core file on a system with dynamic libraries different from those on the system where the core file was made. See the section titled “core-file” in Chapter 7 of the *NightView User’s Guide* (0890395).

5.2. Changes in this Release

The following changes are included in NightView 5.8:

- NightView now allows the type `bool` in expressions when the language is C. (`bool` was already allowed in C++.) See the section titled “Expression Evaluation” in Chapter 3 of the *NightView User’s Guide* (0890395). See the section titled “set-language” in Chapter 7 of the *NightView User’s Guide* (0890395).
- NightView 5.8 has improved diagnostic messages if the **`NightView.p`** program cannot be found, due to the **`Nviewp`** package not being properly installed. See “System Installation” on page 4. See the appendix titled “Implementation Overview” (Appendix E) in the *NightView User’s Guide* (0890395).
- NightView sets the `TERM` enviroment variable to `dumb` in the dialogue shell to avoid problems with some shell programs. See the section titled “Dialogues” in Chapter 3 of the *NightView User’s Guide* (0890395).

NightView 5.8 fixes these problems:

- A very large number of stack frames could cause NightView to abort. NightView 5.8 can handle a much larger number of stack frames. See the section titled “Stack” in Chapter 3 of the *NightView User’s Guide* (0890395).
- Some menus, and some areas of the Monitor Window, did not respond to context-sensitive help. See the section titled “Context-Sensitive Help” in Chapter 3 of the *NightView User’s Guide* (0890395).
- Sometimes text was not properly underlined when searching in the source window in disassembly mode. See the section titled “forward-search” in Chapter 7

of the *NightView User's Guide* (0890395). See the section titled "Debug View Menu" in Chapter 9 of the *NightView User's Guide* (0890395).

- Source directory search didn't work after a **`load`** command. See the sections titled "directory" and "load" in Chapter 7 of the *NightView User's Guide* (0890395).
- Only one process was handled in Data Window dialog boxes. See the section titled "Data Window Dialog Boxes" in Chapter 9 of the *NightView User's Guide* (0890395).
- The shared-memory mailbox unlocking code could possibly let one of the NightView processes see stale data in the other process. See the appendix titled "Implementation Overview" (Appendix E) in the *NightView User's Guide* (0890395).

6.0. Cautions

Special consideration should be given to the following areas:

6.1. SFNOLIM Kernel Tunable

The `SFNOLIM` kernel tunable should not be set above 1024; doing so may cause NightView to fail during initialization or when communicating with other tools.

6.2. Next over Fork

If you **`next`** over a routine that creates a child process and waits for it to finish (e.g., `system()`), the **`next`** command will not complete because the debugger has the child process stopped. You cannot switch to the child process to resume it because you cannot switch away from the parent process while it is doing the **`next`** command. In this case, you need to interrupt the **`next`** operation with the Interrupt button. Then you can switch to the child process and control it.

This does not happen if the debugger does not stop the child process. See the section titled "nodebug" in Chapter 7 of the *NightView User's Guide* (0890395). See the section titled "set-children" in Chapter 7 of the *NightView User's Guide* (0890395).

6.3. Fortran 77 Arrays

There is currently a problem with Fortran 77 arrays passed as formal arguments on PowerMAX OS systems. The compiler generates a temporary variable to optimize references to the array. This temporary variable is then used in place of the original array argument. If the original array argument is not used later in the routine, then the lifetime of the argument is very short and its value is discarded. If you try to use the debugger to reference the argument later in the routine, the results are undefined. You can do two things to avoid running into this problem:

- Use the array later in the routine. For example, pass the array as an argument to another routine. This extends the lifetime of the original argument.
- Use the debugger to look at the array at the first executable line of the routine. The value of the original argument probably still exists at this point.

6.4. Scroll Bars

Some users have had a problem using the scroll bars in the graphical user interface. When the up-arrow button on a scroll bar is clicked, the window scrolls the length of the entire window contents, rather than just one line. This problem is caused by an inappropriate X resource, which has apparently been copied from one user to another in much the same way as **`$HOME/.profile`** files are often copied:

```
*XmScrollBar*translations:   <Btn1Down>:   Select()
```

You should remove this resource from your **`$HOME/.Xdefaults`** or whatever other means you use for setting X resources. This resource affects any Motif application, not just NightView.

6.5. Accessing Variables While in Subprogram Prologue

Because the compilers generate line-number information for code that comprises the prologue of a subprogram, you may inadvertently set an eventpoint or stop your program at such a location. The prologue is the code that the compiler generates to set up the execution environment for a subprogram; until that code completes execution, the environment is incomplete. Attempts to access variables while in the context of the prologue may result in errors from NightView or may result in erroneous values.

If you suspect such a problem, you can determine whether you are in the prologue by doing the following:

- Stop the program at the point where the referencing problem occurs.
- Enter the command **`info frame`**.
- If the output from this command says something like "Could not obtain frame description information", then you are probably stopped in the prologue.

To try and avoid this problem, do not set an eventpoint on the lines containing a subprogram heading. If you want to set the eventpoint at the beginning of the subprogram, use the *unit_name* (for Ada) or *function_name* form of location specifier, rather than a *file_name* and *line_number*. See the section titled "Location Specifiers" in Chapter 7 of the *NightView User's Guide* (0890395).

6.6. Shared Objects

For core files, NightView cannot always find correct stack description information (tdesc) for shared objects. This problem occurs when the shared objects are built without position-independent code. This problem can cause NightView to display the wrong results for any routine other than the most recently called routine. For example, a **`backtrace`** command may print the wrong information, or an **`up`** command may fail. This is true for any release of NightView, not just 5.8. This is not a problem for running processes, only for core files.

We recommend that you build shared objects with position-independent code if you plan to debug programs that use them. For C and Fortran, use the **`-Zpic`** option. See **`cc(1)`** or **`f77(1)`**. For Ada, use **`-sm shared`** or **`-sm both`**. For HAPSE Ada, see "Shared Libraries and Shared-Objects" in the *HAPSE Reference Manual* (0890288). For MAXAda, see "Compile Options" and "Shared Objects" in the *MAXAda Reference Manual* (0890516).

7.0. Direct Software Support

Software support is available from a central source. If you need assistance or information about your system, please contact the Concurrent Software Support Center at 1-800-245-6453. Our customers outside the continental United States can contact us directly at 1-954-283-1822 or 1-305-931-2408. The Software Support Center operates Monday through Friday from 8 a.m. to 7 p.m., Eastern Standard time.

Calling the Software Support Center gives you immediate access to a broad range of skilled personnel and guarantees you a prompt response from the person most qualified to assist you. If you have a question requiring on-site assistance or consultation, the Software Support Center staff will arrange for a field analyst to return your call and schedule a visit.