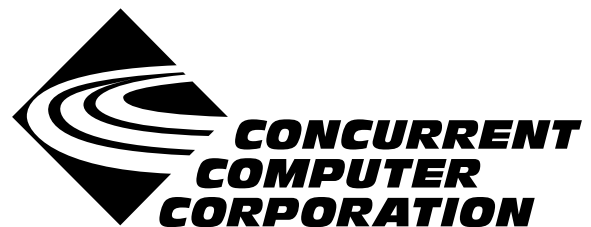


NightView

Version 5.4 Release Notes (PowerMAX)

June 2002

0890395-5.4



Copyright

Copyright 2002 by Concurrent Computer Corporation. All rights reserved. This publication or any part thereof is intended for use with Concurrent Computer Corporation products by Concurrent Computer Corporation personnel, customers, and end-users. It may not be reproduced in any form without the written permission of the publisher.

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

Duplication of this manual without the written consent of Concurrent Computer Corporation is prohibited. Any copy of this manual reproduced with permission must include the Concurrent Computer Corporation copyright notice.

Trademark Acknowledgments

NightView, Power Hawk, PowerMAX OS, PowerMAXION, and TurboHawk are trademarks of Concurrent Computer Corporation.

Night Hawk is a registered trademark of Concurrent Computer Corporation.

Élan License Manager is a trademark of Élan Computer Group, Inc.

PowerStack is a trademark of Motorola, Inc.

Linux is a registered trademark of Linus Torvalds.

X Window System and X are trademarks of The Open Group.

Motif is a registered trademark of The Open Group.

Contents

1.0 Introduction	1
2.0 Documentation	2
3.0 Prerequisites	3
3.1 Software	3
3.2 Hardware	3
4.0 System Installation	4
5.0 Overview of NightView 5.4	5
5.1 Enhancements	5
5.1.1 Data Window	5
5.1.2 Attach Enhancements	5
5.1.3 New Debug Window Buttons	5
5.1.4 Watchpoints	5
5.2 Changes in This Release	6
5.2.1 No Ada License	6
5.2.2 Eventpoints	6
5.2.3 Location Lists	6
5.2.4 Core Files	6
5.2.5 Language Support	6
5.2.6 Remote Debugging	7
5.2.7 Miscellaneous	7
6.0 Cautions	8
6.1 SFNOLIM Kernel Tunable	8
6.2 Breakpoint Command Streams	8
6.3 Fortran 77 Arrays	8
6.4 Scroll Bars	8
6.5 Accessing Variables While in Prologue of Subprogram	8
6.6 Shared Objects	9
7.0 Direct Software Support	10

1.0. Introduction

NightView™ 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.

NightView 5.4 has these new features:

- The Data Window provides a convenient graphical method for browsing data.
- You can attach to a process via a menu and popup showing the processes on the system, or by using an option when you invoke NightView.
- More buttons have been added to the Debug Window.
- The number of watchpoints allowed in a process has changed.

NOTE

Installation procedures for this product are slightly different from those for other Concurrent products. Please read *System Installation* on page 4 of these release notes before attempting to install this product, and carefully follow the installation instructions.

2.0. Documentation

Table 2-1 lists the NightView 5.4 documentation available from Concurrent.

Table 2-1. NightView Version 5.4 Documentation

Manual Name	Pub. Number
<i>NightView User's Guide</i>	0890395-230
<i>NightView Version 5.4 Release Notes (PowerMAX)</i>	0890395-5.4
<i>NightView Pocket Reference</i>	0890475-050

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 5.4 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 5.4 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 XTM server is required if you are using NightView's graphical user interface.
- Élan License ManagerTM 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 and 740

PowerStackTM II and III

Night Hawk[®] Series 6000

TurboHawkTM

PowerMAXIONTM

- 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 5.4

5.1. Enhancements

5.1.1. Data Window

The Data Window lets you browse through your program's data.

You can display:

- variables and other expressions
- all the local variables for the current frame
- registers
- the call stack
- threads

You can access the Data Window with the **Data Display** button or through the **Display** menu in the **Debug Window**.

See the sections titled “Debug Command Buttons”, “Debug Display Menu”, and “Data Window” in Chapter 9 of the *NightView User's Guide* (0890395).

5.1.2. Attach Enhancements

In addition to using the **attach** command, you can now attach to existing processes through the graphical user interface. The **Attach...** button in the **Dialogue** menu of the **Dialogue Window** pops up a dialog that lets you select processes by PID, by user name, and by command name. See the section titled “attach” in Chapter 7 of the *NightView User's Guide* (0890395). See the section titled “Dialogue Menu” in Chapter 9 of the *NightView User's Guide* (0890395).

You can also attach to an existing process with an option when you invoke NightView. See Chapter 9, “Invoking NightView”, in the *NightView User's Guide* (0890395).

5.1.3. New Debug Window Buttons

Up and **Down** buttons let you move within the call stack. The **Data Display** button lets you add an expression to a Data Window. See the section titled “Debug Command Buttons” in Chapter 9 of the *NightView User's Guide* (0890395).

5.1.4. Watchpoints

You can now have more than one watchpoint in a process, as long as no more than one watchpoint is enabled in each process. See the section titled “Watchpoints” in Chapter 3 of the *NightView User's Guide* (0890395).

5.2. Changes in This Release

5.2.1. No Ada License

On systems that require a run-time license, NightView no longer requires a separate Ada license to debug Ada programs. See Chapter 6, “Invoking NightView”, in the *NightView User’s Guide* (0890395).

5.2.2. Eventpoints

NightView 5.4 fixes these problems:

- Monitorpoints did not work for bit-packed Ada types. See the section titled “Monitorpoints” in Chapter 3 of the *NightView User’s Guide* (0890395).
- Monitorpoint field widths were wrong for some expression types.
- NightView would abort if you entered an **mcontrol off** command before the Monitor Window existed. See the section titled “mcontrol” in Chapter 7 of the *NightView User’s Guide* (0890395).
- NightView would abort if you entered an invalid **watchpoint** command. See the section titled “watchpoint” in Chapter 7 of the *NightView User’s Guide* (0890395).
- Setting a breakpoint at the beginning of a function, where the function begins with an inline function call, would set a breakpoint at every instance of that inline. See the section titled “Inline Subprograms” in Chapter 3 of the *NightView User’s Guide* (0890395).
- Attempting to set a tracepoint, while the process was running, when no eventpoints had been set in the process before, would cause NightView to be unable to set any eventpoints in that process. See the section titled “Tracepoints” in Chapter 3 of the *NightView User’s Guide* (0890395).

5.2.3. Location Lists

NightView 5.4 has improved support for location lists generated by the compiler. Location lists give detailed information about the location of variables by code address.

5.2.4. Core Files

info memory is now implemented for core files. See the section titled “info memory” in Chapter 7 of the *NightView User’s Guide* (0890395). NightView no longer incorrectly warns that the stack pointer is out of range for core files.

This release fixes a problem with reading from core files if the read crossed segment boundaries.

5.2.5. Language Support

The default value of **set-overload routine** is now on. See the section titled “set-overload” in Chapter 7 of the *NightView User’s Guide* (0890395).

NightView 5.4 fixes these problems:

- NightView would hang when printing very large arrays in a remote debug session.
- NightView would sometimes show a type name in the local variables list. See the section titled “info locals” in Chapter 7 of the *NightView User’s Guide* (0890395).
- NightView would abort when doing **info locals** on some Ada programs.
- NightView would give an unclear message if you tried to print an expression which was a function returning `void`.

- NightView could not call out-of-line copies of inline member functions in C++.
- NightView would not get the right results comparing an unsigned value and a signed value.
- The performance of printing large arrays is improved.
- Ada enumeration constants are now considered candidates for overloading. See the section titled “Overloading” in Chapter 3 of the *NightView User’s Guide* (0890395).
- Floating-point literals now work when the language is Ada.
- More space is allocated for processing complex expressions.

5.2.6. Remote Debugging

C++ is no longer required to be installed on the target system when you are using remote debugging and you specify a time quantum. See the section titled “Remote Login Dialog Box” in Chapter 9 of the *NightView User’s Guide* (0890395).

NightView 5.4 fixes a problem that would cause NightView to hang when writing large amounts of data between NightView and NightView.p during remote debugging.

5.2.7. Miscellaneous

NightView 5.4 fixes a problem that would sometimes cause NightView to abort on exit.

The `dabr` register is no longer printed by **info registers** by default. See the section titled “info registers” in Chapter 7 of the *NightView User’s Guide* (0890395).

NightView 5.3 would fail to dump out the remainder of the log for **set-log** if you did not do a **set-log close** before exiting NightView. NightView 5.4 dumps out the remainder of the log when it exits.

6.0. Cautions

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. Breakpoint Command Streams

The current implementation of breakpoint command streams does not include protection against concurrent access of data structures internal to the debugger. Avoid referencing other processes within breakpoint command streams. In practice, this is rarely a problem.

6.3. Fortran 77 Arrays

There is currently a problem with Fortran 77 arrays passed as formal arguments. 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 Prologue of Subprogram

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 5.4. 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.

