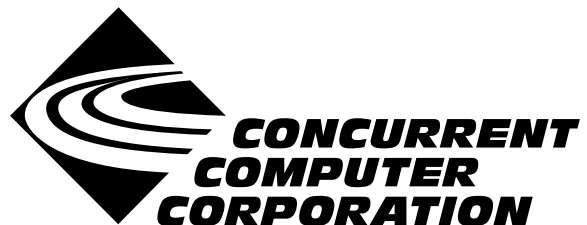


NightView

Version 5.3 Release Notes (PowerMAX)

April 2001

0890395-5.3



Copyright

Copyright 2001 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 5.3	5
5.1 Enhancements	5
5.1.1 Watchpoints	5
5.1.2 Long Long	5
5.1.3 Default Translations	5
5.1.4 Dynamically-linked Programs	5
5.2 Changes in This Release	6
5.2.1 Stack Pointer and Program Counter	6
5.2.2 Graphical User Interface	6
5.2.3 Internal Communications	6
5.2.4 Login Command	6
5.2.5 Vectors	6
5.2.6 Selecting Context	6
5.2.7 Dialogue Output	6
5.2.8 Searching	7
5.2.9 Online Help	7
5.2.10 Finish Command	7
5.2.11 Language Support	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 5.3 has these new features:

- Watchpoints: the process stops when it reads or writes a specified memory location.
- Long long support for C and C++.
- Default object filename translations for remote debugging.

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 5.3 documentation available from Concurrent.

Table 2-1. NightView Version 5.3 Documentation

Manual Name	Pub. Number
<i>NightView User's Guide</i>	0890395-220
<i>NightView Version 5.3 Release Notes (PowerMAX)</i>	0890395-5.3
<i>NightView Pocket Reference</i>	0890475-040

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 5.3 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 5.3 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. 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* (0890395-220).

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 debugging Ada programs, then you need an additional license. The feature alias to debug Ada programs is **NightView_Ada**. 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 5.3

5.1. Enhancements

5.1.1. Watchpoints

A watchpoint stops your program when a particular area of memory is read or written. See the section titled “Watchpoints” in Chapter 3 of the *NightView User's Guide* (0890395-220).

NightView allows you to specify variables or specific memory locations to watch for reads or writes, using hardware features to minimize intrusion on the process. The process stops when the specified location is referenced or is changed, depending on the watchpoint options. Like other eventpoints, watchpoints can be enabled or disabled and can have conditions and ignore counts.

5.1.2. Long Long

NightView now supports the 64-bit integer types available in C and C++ as `long long` and `unsigned long long`.

5.1.3. Default Translations

NightView 5.3 has default object filename translations. This is interesting only for remote debugging. See the section titled “Remote Debugging” in Chapter 3 of the *NightView User's Guide* (0890395-220).

Filename translations are required when an executable program or shared library isn't accessible via the same filename from the local and remote systems. An example where filename translation is useful is when you are debugging PowerMAX OS programs that use shared libraries from a Linux host system. For example, the target PowerMAX OS executable might require `/usr/lib/libc.so.1` which is accessible from the host Linux system as `/pmax/os/4.3/moto/usr/lib/libc.so.1`.

The default translations are formed by examining the system mount table and by detecting which cross development environments exist on the local system. In many cases, the default translations are sufficient, so the user does not need to use the `translate-object-filename` command. See the section titled "translate-object-filename" in Chapter 7 of the *NightView User's Guide* (0890395-220).

Earlier releases of NightView used the first translation that matched the object filename. NightView 5.3 goes through the matching translations until it finds a file containing a text segment with the same contents as the text segment of the program executing. If no file is found with the same text segment contents, then NightView gives a warning and uses the first translation that matched the object filename.

5.1.4. Dynamically-linked Programs

If your program is dynamically-linked, NightView runs the process just far enough to complete the dynamic linking. The final step of dynamic linking is to call any initialization routines in each dynamic library. Earlier versions of NightView stopped the process after these initialization routines were complete. NightView 5.3 stops the process just before these routines are called. This allows you to debug the initialization routines. See the section titled “Debugging with Shared Libraries” in Chapter 3 of the *NightView User's Guide* (0890395-220).

5.2. Changes in This Release

5.2.1. Stack Pointer and Program Counter

NightView now does rudimentary checks of the stack pointer and program counter each time the program stops. NightView gives a warning if either one contains an address which is not part of the process or does not have the appropriate permissions.

5.2.2. Graphical User Interface

NightView 5.3 improves the performance of removing monitorpoints in the graphical user interface.

NightView 5.3 fixes a problem that would sometimes cause the X library to get a "Bad drawable" error with monitorpoints.

NightView 5.3 fixes a problem with opening a second debug window on Linux.

NightView 5.3 fixes a problem with trying to remove the commands attached to a breakpoint from the GUI.

5.2.3. Internal Communications

For a local dialogue, NightView uses shared memory to communicate between the two processes that make up NightView. NightView 5.3 makes these communications more reliable on systems running programs at a high priority.

5.2.4. Login Command

If a **login** command fails to connect to the remote system, the command terminates a non-interactive command stream. See the section titled "Command Streams" in Chapter 3 and the section titled "login" in Chapter 7 of the *NightView User's Guide* (0890395-220).

5.2.5. Vectors

When printing a vector with the **print** command, you can now use the **/f** option to print the vector as 4 floating point numbers. All other formats print as 4 hexadecimal numbers.

The **vector-set** command now works correctly on Linux. See the section titled "vector-set" in Chapter 7 of the *NightView User's Guide* (0890395-220).

5.2.6. Selecting Context

NightView 5.2 could not select C threads when running on PowerMAX OS 4.3. NightView 5.3 can select C threads on either PowerMAX OS 4.3 or PowerMAX OS 5.1. See the section titled "select-context" in Chapter 7 of the *NightView User's Guide* (0890395-220).

The **select-context** command now works correctly for Ada tasks on Linux.

5.2.7. Dialogue Output

NightView 5.3 fixes a problem where nulls in the dialogue output would truncate the rest of the line.

5.2.8. Searching

NightView 5.3 changes how **forward-search** and **reverse-search** decide where to begin searching. See the sections titled "forward-search" and "reverse-search" in Chapter 7 of the *NightView User's Guide* (0890395-220).

5.2.9. Online Help

NightView 5.3 improves the wording of some of the help pages for errors and adds more cross references.

The command **help info** takes you to a page with links to all the **info** commands.

In earlier releases, you could get help on an individual command by entering the command **help command**. Now, if the command has a special abbreviation, you can also get help on the command by its abbreviation: **help abbreviation**. For example, the **backtrace** command has a special abbreviation **bt**, so you can get help on the **backtrace** command with **help bt**. See the sections titled "Command Syntax" and "help" in Chapter 7 of the *NightView User's Guide* (0890395-220).

5.2.10. Finish Command

The **finish** command now works correctly for routines called by a patchpoint. See the section titled "finish" in Chapter 7 of the *NightView User's Guide* (0890395-220).

5.2.11. Language Support

In earlier releases, NightView could sometimes fail to recognize that a function had only inline instances and would let you try to call the function. Calling functions with only inline instances is now disallowed.

In addition, NightView 5.3 fixes various problems in the following areas:

- comparing Ada bit-packed arrays
- subscripting arrays of records in Ada programs
- referring to class members in `dynamic_cast<>()` expressions
- casting to an enum type
- referring to types nested in classes
- the scoping of Ada renames

NightView 5.3 also fixes a problem that would sometimes cause enums to show duplicate values.

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* (0890395-220).

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 5.3. 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.

