

# Sun HPC ClusterTools™ 3.1 Product Notes

---



THE NETWORK IS THE COMPUTER™

**Sun Microsystems, Inc.**  
901 San Antonio Road  
Palo Alto, CA 94303-4900 USA  
650 960-1300 Fax 650 969-9131

Part No. 806-4182-10  
March 2000, [Revision A](#)

Send comments about this document to: [docfeedback@sun.com](mailto:docfeedback@sun.com)

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd. For Netscape Communicator™, the following notice applies: (c) Copyright 1995 Netscape Communications Corporation. All rights reserved.

Sun, Sun Microsystems, the Sun logo, SunStore, AnswerBook2, docs.sun.com, Solaris, Sun HPC ClusterTools, Prism, Sun Performance WorkShop Fortran, Sun Performance Library, Sun WorkShop Compilers C, Sun WorkShop Compilers C++, Sun WorkShop Compilers Fortran, Sun Visual WorkShop, and UltraSPARC are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

**RESTRICTED RIGHTS:** Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd. La notice suivante est applicable à Netscape Communicator™: (c) Copyright 1995 Netscape Communications Corporation. Tous droits réservés.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, Solaris, Sun HPC ClusterTools, Prism, Sun Performance WorkShop Fortran, Sun Performance Library, Sun WorkShop Compilers C, Sun WorkShop Compilers C++, Sun WorkShop Compilers Fortran, Sun Visual WorkShop, et UltraSPARC sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPENDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



# Contents

---

Related Software	1
Solaris Versions Supported	1
Sun Software Patches	2
Early Access Sun WorkShop™ 6.0	3
Load Sharing Facility Suite	4
What's New in Version 3.1	5
ClusterTools Installation	5
ClusterTools Documentation	5
Sun SCI/RSM	6
Sun MPI	6
Sun MPI I/O	6
Sun CRE	7
PFS	7
Prism Environment	8
Bugs and Caveats in Version 3.1	9
ClusterTools Installation	9
ClusterTools Documentation	10
Sun SCI/RSM	11
Sun MPI	13

Sun MPI I/O	14
Sun S3L	14
Sun CRE	15
PFS Administration	15
Prism Environment	16
Prism Environment: General	16
Prism Performance Analysis	19
Prism Environment: Fortran and C Programs	21
Prism Environment: C Programs	22
Prism Environment: Fortran Programs	22

# Sun HPC ClusterTools 3.1 Product Notes

---

Sun HPC ClusterTools 3.1 is a follow-on to the 3.0 release, designed to support the latest release of the Solaris™ operating environment and to enhance the ClusterTools functionality.

These product notes report:

- User-visible changes in the ClusterTools product since Version 3.0, beginning on page 5
- Bugs and caveats in the various ClusterTools components, beginning on page 9

---

## Related Software

### Solaris Versions Supported

Sun HPC ClusterTools 3.1 can be used with the Solaris 2.6, Solaris 7, and Solaris 8 operating environments (32-bit and 64-bit).

# Sun Software Patches

The following three sets of patches are provided on the ClusterTools 3.1 CD.

## ■ Solaris 2.6 Operating Environment

These patches are required on clusters running the Solaris 2.6 operating environment in order to use the functionality noted.

Patch No.	Description	Required for
105181-19	Kernel update patch	Basic ClusterTools functionality, including the CRE and MPI
105210-25	libaio, libc, and watchmalloc patch	Basic ClusterTools functionality, including the CRE and MPI
105568-14	/usr/lib/libthread.so.1 patch	Basic ClusterTools functionality, including the CRE and MPI
105490-07	Linker patch	Building executable programs
105591-07	Shared library patch for C++	Using the Prism debugger or building C++ MPI programs

## ■ Solaris 7 Operating Environment

These patches are required on clusters running the Solaris 7 operating environment in order to use the functionality noted.

Patch No.	Description	Required for
106541-08	Kernel update patch	Basic ClusterTools functionality, including the CRE and MPI
106980-07	libthread patch	Basic ClusterTools functionality, including the CRE and MPI
106950-06	Linker patch	Building executable programs
106327-05	Shared library patch for C++	Using the Prism debugger or building C++ MPI programs
106300-06	Shared library patch for 64-bit C++	Using the Prism debugger or building 64-bit C++ MPI programs
107838-01	libtnfctl patch	Using the Prism debugger

## ■ Sun WorkShop™ 5.0 Compilers

These patches are provided for convenience. Without them, you may find some performance degradation, and some programs (notably Fortran 90 and C++) may not execute properly.

Patch No.	Description
107355-04	WorkShop IPE 5.0: Patch for dbx
107357-05	Compiler Common 5.0: Patch C 5.0, C++ 5.0, F77 5.0, F90 compilers
107358-04	WorkShop IPE 5.0: Patch for 64-bit dbx
107390-07	C++ 5.0: Patch for 64-bit C++ 5.0 compiler
107596-03	F77 5.0: Patch for FORTRAN 77 (f77) 5.0 compiler
107989-01	Fortran Common 5.0: Patch F77 5.0, F90 2.0 compilers

## Early Access Sun WorkShop™ 6.0

The ClusterTools 3.1 software components are expected to work with early access (EA) versions of Sun WorkShop 6.0, with the following known exceptions:

- The Prism environment may not reliably recognize symbol table entries.
- The Prism environment cannot be used to debug C++ programs.
- Missing functions in `libsunperf` prevent Sun S3L from working with 64-bit Sun WorkShop 6.0 EA4 or EA5. Failure occurs at link time, and no executable is produced.

Sun S3L is known to work with 32-bit Sun WorkShop 6.0 EA4.

A patch is planned for ClusterTools 3.1 to ensure full compatibility with Sun WorkShop 6.0.

# Load Sharing Facility Suite

Users may choose to run Sun HPC ClusterTools 3.1 with the Load Sharing Facility (LSF) suite from Platform Computing as resource manager, as an alternative to the CRE.

- Those who previously purchased LSF 3.2.3 through Sun Microsystems must install the following patches to update the LSF suite to support Sun HPC ClusterTools 3.1:

- 108691-01 LSF daemons and libs updates and fixes
- 108692-01 LSF user and examples updates and fixes

Contact Sun Service to obtain these patches.

- Those who purchase LSF software directly from Platform Computing should contact Platform to obtain the proper LSF version to support Sun HPC ClusterTools 3.1.



---

# What's New in Version 3.1

This section provides a checklist of the user-visible differences between Versions 3.0 and 3.1 of Sun HPC ClusterTools, organized by component.

## ClusterTools Installation

- ClusterTools Versions 3.0 and 3.1 can coexist on the nodes of a cluster. This includes the following support:
  - The administrator specifies which ClusterTools version, 3.0 or 3.1, is active, either via the installation tool `install_gui` or with a command entered at a shell prompt.
  - The administrator can selectively remove a ClusterTools version, either via new `install_gui` controls or with a command entered at a shell prompt.
- SCI installation is now separate from the ClusterTools installation process; it is accomplished using a command at the shell prompt

## ClusterTools Documentation

- ClusterTools documents and other select Sun technical documents can now be purchased from Fatbrain.com, an Internet professional bookstore. Visit them at:

<http://www1.fatbrain.com/documentation/sun>

As before, you can also access ClusterTools documents and other Sun technical documents on the Web at:

<http://docs.sun.com>

- Version 3.1 includes a new user manual, *Sun HPC ClusterTools 3.1 Performance Guide*, which provides tips for getting the best performance from Sun MPI programs.
- A new manual, *Sun HPC ClusterTools 3.1 User's Guide*, expands and supersedes the *Sun MPI User's Guide* that accompanied earlier releases.

## Sun SCI/RSM

- Administrator control (via the `hpc.conf` file) of LSF batch queue access to particular SCI interfaces for RSM communication

## Sun MPI

- Scalability to 64 nodes and 1024 processes
- MPI-2 dynamic support and limited support for on-node one-sided communication, including these new routines:

```
MPI_Accumulate, MPI_Alloc_mem, MPI_Free_mem, MPI_Get,  
MPI_Put, MPI_Win_c2f, MPI_Win_create,  
MPI_Win_create_errhandler, MPI_Win_create_keyval,  
MPI_Win_delete_attr, MPI_Win_f2c, MPI_Win_fence,  
MPI_Win_free, MPI_Win_free_keyval, MPI_Win_get_attr,  
MPI_Win_get_errhandler, MPI_Win_get_group, MPI_Win_get_name,  
MPI_Win_lock, MPI_Win_set_attr, MPI_Win_set_errhandler,  
MPI_Win_set_name, MPI_Win_unlock
```

## Sun MPI I/O

- Reduced overhead on shared offset operations on large configurations
- Reduced overhead for opening files
- Environment variables corresponding to some of the settable `MPI_Info` hints
- New `MPI_Info` hint, `mpio_concurrency`, for controlling the level of underlying thread concurrency used by nonblocking routines

## Sun CRE

- Scalability to 64 nodes and 1024 processes
- Coexistence of Version 3.1 with Version 3.0
- New `mprun` features:
  - `-zt` for rank tiling
  - `-Mf` for rankmap specification
  - `-u` to allow inclusion of “independent” nodes
  - Flag matrix for determining semantics of `mprun` requests
- The compilation utilities beginning with `tm`, such as `tmcc`, are now available with `mp`, such as `mpcc`.
- Faster startup times for jobs
- Kerberos Version 4 authentication no longer supported

## PFS

- A new kernel module to be loaded on nodes running the Solaris 8 operating environment
- Support for Kerberos Version 5 authentication; Kerberos Version 4 no longer supported
- An expanded set of configuration variables in the `hpc.conf` file for PFS disks and servers, including:
  - `threads=`
  - `auth=`
  - `servcong=` for controlling server-side congestion on heavily loaded networks
  - `clntcong=` for controlling client-side congestion on heavily loaded networks

---

**Note** – The `pfs.ioc` command-line options `-t` (for threads) and `-a` (for authorization method) are deprecated in Version 3.1. This information should be supplied in `hpc.conf` instead.

---

# Prism Environment

The Prism environment's pset syntax has been extended to support debugging multiprocess, multithreaded programs. The new features are:

- New commands
  - thread
  - threads
  - lwps
  - sync
  - syncs
- New user interface features
  - The main Prism window allows you to cycle among the entries in a list of psets.
  - The Psets window represents the threads in each process as vertical stripes.
  - The Where graph represents the stack traces of threaded programs as a forest of trees.
- New command line options
  - `-threads | -nothreads` option
  - `-install` option (specifies a private colormap at Prism startup)

---

# Bugs and Caveats in Version 3.1

## ClusterTools Installation

- Advice for setting `/etc/system` parameters is out of date. [no associated bug ID]

Table 2-1 of the *Sun HPC ClusterTools Installation Guide* contains advice for setting various system parameters in `/etc/system`. Only the `pt_cnt` recommendation is valid. All other parameter-setting recommendations given in Table 2-1 are based on outdated information. The `/etc/system` portion of Table 2-1 should read as follows:

---

<code>/etc/system</code> file settings	Add the following entry to the <code>/etc/system</code> file on each node in your cluster:
--	--

```
set pt_cnt=1024
```

After making the change, reboot each node. Changes to parameter values made in an `/etc/system` file will not take effect until the node is rebooted.

---

- Redundant prompts when using `rsh` method for an NFS installation in a ClusterTools 3.0/3.1 environment [4302672]

If you use the `rsh` method to install ClusterTools 3.1 on an NFS server that also has ClusterTools 3.0 installed, you will be notified by each node that the daemons are about to be shut down and asked if you want to continue. If you want the installation to proceed, just answer yes to each prompt.

- During NFS installation, avoid setting `INSTALL_LOC` to `/opt/SUNWhpc`. [4240690]

During an NFS installation, if the client mount point (`INSTALL_LOC`) `hpc_config` is set to `/opt/SUNWhpc`, the installation will fail.

# ClusterTools Documentation

- Instructions are not provided for installing the Sun HPC ClusterTools 3.1 AnswerBook2 document collection. [no associated bug ID]

You must be logged in as root for the following procedure. Also, the ClusterTools package containing the AnswerBook2 files must be installed on a host that is set up to be an AnswerBook2 server.

---

**Note** – If you don't already have a machine configured as an AnswerBook2 server, see the README files that are on the Solaris Documentation CD. They contain instructions for setting up an AnswerBook2 server.

---

1. Log in as root on the AnswerBook2 server.
2. Insert the CD into your CD-ROM drive. The Volume Manager should automatically mount the CD.
3. Go to the location of the document collection packages.

```
# cd /cdrom/hpc_3_1_ct/Product
```

4. Use the `pkgadd` utility to install the `SUNWabhpc` package:

```
# pkgadd -d . SUNWabhpc
```

5. You will be asked whether you want to install all AnswerBook2 files in `SUNWabhpc` (if so, choose `heavy`) or to install only the essential files (if so, choose `nil`).

Selecting `heavy` causes all files to be copied to your server system. This will consume close to 6.3 megabytes of disk space, but does provide the fastest server performance.

6. Specify the parent path for the AnswerBook2 collection directory. For example, `/opt`.
7. AnswerBook2 collection packages should include a post-installation script that adds the collections to the server's database and restarts the server. If this does not happen, use either the AnswerBook2 Administration page's Add Collection to List function or the `ab2admin -o add_coll` command to add these collections to the documentation server's database.

If you have other document collections already installed on your system, you can use either the AnswerBook2 Administration page's Scan for Locally Installed Collections function or the `ab2admin -o scan` command to have them found and added to the documentation server's database. This function only works for locally installed packages.

To serve document collections located on some other system, use the Add Collection to List function or the `ab2admin -o add_coll` command and provide the full path name to the `ab_cardcatalog` or `collinfo` file. For example, if the documentation server is named `imaserver` and you want it to serve document collections located on the system elsewhere, you might use the following command:

```
# ab2admin -o add_coll -d /net/elsewhere/books/SUNWdtad/collinfo
```

- **Restriction on using `catman` with Solaris 7 or 8 [4300280, 4288174]**

When you run `catman` on man pages in the Solaris 7 or 8 operating environment, “shadow” or “pointer” files are mishandled in the creation of the `windex` database (see `catman(1M)`). The Sun MPI and Sun S3L man page collections include many such shadow files. After running `catman`, the `man(1)` command will do a `windex` lookup by default and, therefore, will fail to find these shadow files. You can work around this problem by using the `-F` option to the `man` command, which bypasses the `windex` file.

## Sun SCI/RSM

- **No man page for RSM daemon [no associated bug ID]**

There is no man page for the RSM daemon, `hpc_rsmd`. This product note is provided as a temporary substitute.

`sunhpc.hpc_rsmd` is a shell script used to start and stop `hpc_rsmd`, the Sun HPC daemon that provides SCI-based Remote Shared Memory (RSM) resources to Sun MPI jobs running on appropriately configured clusters.

`sunhpc.hpc_rsmd` is found in `/etc/init.d` and `/etc/rc3.d`. It follows the conventions of other scripts in those areas. That is, `hpc_rsmd` can be started by invoking `sunhpc.hpc_rsmd` with a `start` parameter and stopped by invoking `sunhpc.hpc_rsmd` with a `stop` parameter. A `clean` parameter is also supported. The `clean` parameter is a synonym for `stop`; it is provided as a mnemonic to be used when an abnormal `hpc_rsmd` exit leaves stale files behind.

Invoking `sunhpc.hpc_rsmd` with either `stop` or `clean` will remove RSM backing store files in `/var/hpc/rsm`, Sun HPC private shared memory files `/tmp/.hpcshm_acf.*.32790` and `/tmp/.hpcshm_mmap.*.32790`, and the `hpc_rsmd` lock file `/tmp/.hpc_rsmd_lock`.

A start invocation will fail if the script does not find evidence of installed SCI drivers, RSM drivers, the RSM configuration file `/etc/sma.config`, or the `hpc_rsmd` binary.

- **Meaningless RSM warning messages on initial RSM package installation [no associated bug ID]**

When the SCI and RSM packages are installed on a Sun HPC cluster for the first time, the following warning message will be displayed on the console:

```
NOTICE: smak: _init: no SCI adapters found usable by SMA
RSM: loading driver.
RSM: maximum exportable memory: 60579840 bytes.
WARNING: RSM: no SCI controllers detected.
```

The following related warning message will be logged in `/var/adm/messages`.

```
doing add_drv on rsm driver...
drvconfig: Driver (rsm) failed to attach
Warning: Driver (rsm) successfully added to system but failed to
attach.
```

Ignore both messages. They are erroneously generated because the RSM packages are installed before the SCI adapters have been initialized. Once the nodes have been rebooted, the RSM's dependencies on the SCI drivers will be satisfied and the artificial condition reported by these messages will be gone.

- **Unnecessary RSM warning message when a node boots up [no associated bug ID]**

When a node that has the RSM packages installed boots up, the following warning message will be displayed on the console:

```
WARNING: RSM: mod_remove failed with error 16.
```

This happens because the RSM daemon is already running when the RSM kernel agent tries to perform the customary load/unload sequence at boot time. Since the RSM daemon is running, the unload step fails.



# Sun MPI

- Signals generated when using `MPI_Get_elements` and `MPI_Status_set_elements` [4311633]

If you are using `MPI_Get_elements` and `MPI_Status_set_elements` when using a 64-bit library, your program may signal when the type in question is an indexed block.

- Possible deadlock when cancelling nonblocking sends over TCP [4246641]

When running over a TCP network using lazy connections, a program that cancels a nonblocking send before the connection to the destination has been established may deadlock.

You can avoid this deadlock by ensuring that all connections are established during initialization. To do this, set the `MPI_FULLCONNINIT` environment variable to 1. For more information about environment variables, see the *Sun HPC ClusterTools Performance Guide* and the *Sun MPI Programming and Reference Guide*.

- Scalability limitations with RSM over SCI [4256037, 4256547]

Your Sun MPI program may fail in `MPI_Init` if you run a large number of processes using Remote Shared Memory (RSM) over a Scalable Coherent Interface (SCI) network. If this happens, you will see an error message similar to this:

```
[1043 MPI_COMM_WORLD 89] ERROR in MPI_Send: internal error:
rsm_create_sendpools: cannot open rsm controller
Aborting.
[1043 MPI_COMM_WORLD 93] ERROR in MPI_Send: internal error:
rsm_create_sendpools: cannot open rsm controller
Aborting.
```

The maximum number of processes that can be run over RSM is a per-node limitation and depends on the number of nodes, the number of controllers per node, and the striping level used by each process when establishing an RSM connection.

To avoid this failure, the number of processes that can be spawned on each node that uses RSM for communication is limited to

$$(255 - C) / (S * (N-1))$$

where  $C$  is the number of controllers per node,  $S$  is the striping level, and  $N$  is the number of nodes in the cluster.

This limitation is due to a limitation in the SCI driver.

## Sun MPI I/O

- Erroneous error codes generated when using `MPI_File_close` and `MPI_File_sync` [4311456]

In the Solaris 8 operating environment, when multiple processes in the same job simultaneously call `MPI_File_close` or `MPI_File_sync`, all but one process may erroneously return the error code `MPI_ERR_UNKNOWN`.

- Silent error when using `MPI_File_seek_shared` [4311382]

A call to `MPI_File_seek_shared` erroneously succeeds when (whence = `MPI_SEEK_END`) and the offset is a negative number large enough to seek to an absolute offset less than 0.

- `MPI_File_sync` not implemented as a collective routine [4310961]

- Failure when using more than 200 processes with PFS [4312938]

When running jobs with more than 200 processes, if the processes access a PFS file, a subset of the processes may hang in the call to `MPI_Finalize`.

## Sun S3L

- Programs linked with `libsunperf` and `libthread` may hang. [no associated bug ID]

When multiple programs linked with `libsunperf` and `libthread` start at the same time, programs sometimes hang. This bug is easily identified; running `pstack` on the hung process will reveal all its threads as stuck in calls to `vfork()`. Other processes will be waiting in `MPI_Init()`. This problem is associated with version 4.x of the Sun Performance Workshop™.

## Sun CRE

- Caveat on using `mpkill` with `-C` option [4303000]

The `-C` option to `mpkill` must be used exactly as documented, that is, as root and *only* from the master node. If you run `mpkill -C` as root from another node in the cluster, the `tm.spm` daemon may fail. If this happens, see the troubleshooting chapter in the *Sun HPC ClusterTools Administrator's Guide* for help with system recovery.

- DES requires `krb5.conf` file [4307072]

DES authentication requires that a file named `/etc/krb5/krb5.conf` be installed. This file need not have any content.

- Kerberos 5 not available [4317489, 4291039]

Kerberos Version 5 authentication is available on the CRE only to root; it is not available to users. In addition, Kerberos 5 on the CRE does not survive ticket renewal.

- Caveat: CCM tool `cconsole` restricts console access [no associated bug ID]

When using `cconsole`, remember that it gives exclusive access to the console device, whereas the other CCM tools, `ctelnet` and `crlogin`, permit multiple users at a time.

## PFS Administration

- *Administrator's Guide* uses wrong filename in Kerberos 5 setup [4316627]

In describing the setup of PFS authentication with Kerberos Version 5, the *Sun HPC ClusterTools Administrator's Guide* refers to the `srvtab` file. This filename has been changed to `keytab` in Kerberos 5.

- Kerberos Version 5 for PFS not usable on systems where the CRE is the resource manager [4317489, 4291039]

# Prism Environment

## Prism Environment: General

- Large Prism jobs may hang on LSF. [4240458]

When the LSF system is running under an extreme load, large Prism jobs may hang, then fail, issuing the following message:

```
Error: Failed to start prism on the nodes. Could not  
determine process ids
```

- The `make` command can cause the commands-only Prism environment to crash. [4309986]

The `make` command can cause the commands-only Prism environment to crash and dump an infinite stack trace. (The `make` command does work properly in the graphical Prism environment.)

- The `print` command sometimes requires fully qualified variable names. [4291488]

To print a variable that is not visible from the current process, the variable needs to be fully qualified.

For example, this command may fail if `x` is not visible from the current process, process 0:

```
(prism all) print x pset 1
```

To work around, use the fully qualified name for `x`. For example:

```
(prism all) print `f1`x pset 1
```

- Function calls change the visiting frame. [4316281]

In the MP Prism environment, calling a function after using an `up` or `down` command (which changes the current frame on the call stack), resets the position on the current stack to the top of the stack. If you want to continue inspecting the function frame as it was before issuing the function call, you must traverse the stack again using `up` or `down` command.

- The graphical Prism environment may crash when running on Solaris 8 and displaying output on Solaris 7 (using CDE). [4312413, 4312815]

In some cases, the 64-bit Prism graphical environment running on Solaris 8 (as a client) may not correctly interact with a CDE running on Solaris 7. An error relating to the `FontSet X` resource will be displayed. To work around this limitation, add the following lines to your `.Xdefaults` file (or uncomment the following lines in the Prism `app-default` file):

```
Prism.textFont: -*-medium-r-*-12-*-*-m*
Prism.masterFont: -adobe-helvetica-bold-r-normal-*-14-140-*-*-iso8859-1
```

Note: you can add custom fonts. The problem occurs with `-dt-interface` style fonts.

- Problems encountered debugging programs compiled without the `-g` option [4313613]

The Prism environment may not display the correct assembly file when debugging applications that have not been compiled with `-g`. Also, the Prism environment may crash when trying to display the assembly file for 64-bit applications not compiled with `-g`.

- Limitation on specifying queues in the LSF environment [4308110]

The `-q` option to the Prism environment (using LSF) does not work. To work around, use the `-bsubargs` option to specify the queue

- The thread `-blocks` command fails in 64-bit Solaris environments. [4282102, 4174129]

The thread `-blocks` command fails in a 64-bit Solaris environment with the following error message:

```
prism: debugger service failed
prism: cannot print locks -- debugger service failed
```

- Linking libraries with relative paths can cause MP Prism assertion errors. [4308152]

MP Prism fails on applications that are linked with libraries using a relative path (such as `-L. /libpath`). To work around, use an absolute path.

- Pset expressions may be evaluated repeatedly. [4232414]

When a pset has been specified using an expression such as

```
pset {foo() != 0}
```

the Prism environment evaluates the expression repeatedly. Thus, in the above example, the function `foo` is called several times. Do not rely on a unique evaluation of such expressions.

- Events on `_start` not detected. [no associated bug ID]

When debugging with the MP mode of the Prism environment, an event (such as a breakpoint) on `_start` will go unnoticed.

There is a workaround. To `stopi` at `_start`, set the event at the instruction that follows `_start`. Use the `print` command to find the address of the next instruction by adding four bytes to the address of `_start`. For example:

```
(prism all) print &_start+4  
0x1078  
(prism all) stopi at 0x1078
```

- Caveat: Ctrl-C replaces Ctrl-Z in `fg` command.

The commands-only mode of the Prism environment prints a message instructing you to use the `fg` command to redirect keyboard input to the debugging process. The `fg` command indicates that you should type Ctrl-C to redirect keyboard input to the Prism environment. (Formerly, the Prism environment accepted Ctrl-Z.)

For example:

```
hpc-450-4 [195]% prism -C -n 1 kbio  
kbio: pid is 12342  
(use the fg command to direct keyboard input to your program)  
(prism all) fg  
(type ctrl-C to return to prism)  
kbio ==>
```

- Harmless warnings from alternate paths to source file [4204717]

Symbolic links allow source files to be reached from several paths. If such a file is compiled from one path and then used in the Prism environment via the `use list` and another path, then the Prism environment may issue a warning that the file does not exist. The warning may be safely ignored.

- Stopping the Prism graphical environment with Ctrl-Z fails. [4226954]

Stopping the Prism graphical environment by using Ctrl-Z and then continuing in the background (using `bg`) does not work. You can launch the Prism environment in the background from the command line by appending `&` (ampersand) to the `prism` command.

## Prism Performance Analysis

- Background pane of `tnfview` panner intentionally left blank [no associated bug ID]

To improve `tnfview`'s performance when profiling large data sets, the `tnfview` panner window has been modified. The panner no longer draws events in its background pane—it draws only a rectangle against a blank background.

- MPI performance analysis maximum of 300 processes [4312658]

The Prism environment currently limits MPI performance analysis to jobs of no more than 300 processes.

- Limitation on performance analysis of 32-bit binaries on Solaris 7 [4222058]

With the application of patch 107838-1 (or later), the Prism environment works with both 64-bit and 32-bit binaries on Solaris 7. However, it cannot do performance analysis of 32-bit binaries on Solaris 7 unless you use the `-32` option with the 32-bit program.

- Several performance analysis commands require the `run` command as a precondition. [4208529]

The Prism `tnflist`, `tnfdebug`, and `tnfdisable` commands require that the Prism `run` command be issued first. For example, if you set a breakpoint at `main` and then issue the `run` command, you can issue the `tnflist` command once the Prism environment halts at `main`.

```
(prism all) tnfcollection on
(prism all) stop in main
(prism all) run
(prism all) tnflist *
```

- MPI performance analysis and disk space [no associated bug ID]

The Prism environment's MPI performance analysis generates large volumes of data in `/usr/tmp`, particularly for long-running programs or programs with high process counts. As a result, performance analysis may fail if insufficient disk space is available in `/usr/tmp`.

The Prism environment uses space in `/usr/tmp` to store the temporary data generated by the TNF probes. You specify the amount of the storage required by performance analysis trace buffer files with the `size` parameter of the `tnffile` command (the `tnfcollection on` command automatically creates trace buffer files, using the `tnffile` default size of 128 KBytes when it creates them).

If your trace buffer files are too small, once the buffer fills up your data will begin to overwrite older data in the trace buffer. If your trace buffer files exceed the size of your target directory (`/usr/tmp`, by default), the data collection process will fail at that stage, before creating the final data file required by `tnfview`. When you have limited space available in `/usr/tmp`, you can shorten the collection time using the `tnfcollection` command or limit the types of events collected using the `tnfenable` command. You can also specify a different location for the trace buffer files by setting the environment variable `PRISM_TNFDIR`. For example,

```
% setenv PRISM_TNFDIR /home/smith/tmp
```

---

**Note** – If you set `PRISM_TNFDIR` to an NFS-mounted directory, your performance analysis data will be affected by the extra time required for writing the data to non-local directories.

---



## Prism Environment: Fortran and C Programs

- The Prism environment cannot pass array arguments. [4189552]

The Prism environment cannot pass array arguments to functions or procedures in Fortran or C programs.

- The Prism environment cannot debug programs compiled with the `-xpg` option of the 4.2 compilers. [4252204]

Because of a bug in the Sun WorkShop 4.2 compilers, the Prism environment cannot run and debug programs that have been compiled with the `-xpg` option.

- S3L array handles sometimes not recognized in events [4281749]

The Prism `type` command affects only its parameter. If several `COMMON` blocks in your program declare the same S3L handle with several names, you may have to issue the `type` command for each name that you want to view as an S3L array. Note that such variables should be visible, as described in the *Prism User's Guide*.

- Assignment to S3L arrays indexed with parallel variables may be unreliable. [4278698]

The Prism environment does not fully support assignments to S3L arrays using parallel indices. Thus, if `A` is a parallel array (that is, an S3L array descriptor after issuing a `type` command) and `x` and `y` are parallel variables, then assignments such as `A[x] = y` may have unexpected results.

- Improper error handling when S3L arrays are indexed with out-of-range indices [4312805]

If out-of-range indices are used to index S3L arrays, the Prism environment may hang or display other unexpected behavior (such as causing some of the debugees to terminate).

## Prism Environment: C Programs

- float arguments passed incorrectly [4171235]

The Prism environment passes the wrong value for `float` arguments in old-style functions in C. For example, if a C program contains the following,

```
int foo(f)
float f;
{...}
```

the Prism environment will not pass the `float` argument correctly to calls of the function `foo`. As a result, expressions such as `print f+foo(f)` may yield incorrect results.

- Wrong value printed for long double variable. [4129556]

The Prism environment does not support the `long double` type.

## Prism Environment: Fortran Programs

- Caveat: Distinguishing `MAIN` from `main` (no associated bug ID)

There is a change in the way the Prism environment handles the symbol `MAIN`. In a Fortran program, the Prism environment used to treat `main` as a synonym for `MAIN`, but only for certain commands, such as `stop in`. Now the Prism environment uniformly treats `main` (the program's entry point) and `MAIN` (the main subroutine of the Fortran program) as separate and distinct entities. If you were previously accustomed to typing `stop in main` meaning `stop in MAIN`, you will now have to type `stop in MAIN`.

- Accessing array members of structures [4187233]

If `arr` is an array member of a structure `s`, then you can access element 1 of `arr` in the Prism environment using an additional pair of parentheses:

```
(prism all) print (s%arr)(1)
```

The Prism environment does not currently support the syntax:

```
(prism all) print s%arr(1)
```

- Pointers to array sections in Fortran 90 handled incorrectly. [no associated bug ID]

The Prism environment does not handle Fortran 90 pointers to array sections correctly. For example, assume that the program is stopped after this statement:

```
array_ptr => some_array(1:10:3)
```

If you try to print `array_ptr`, the Prism environment will print an incorrect assortment of elements and an incorrect number of elements.

- Fortran array assignments limited to continuous strides [4187233]

The Prism `assign` command supports only unit strides in array section assignments. For example:

```
(prism all) assign arr(2:7) = 27
```

is allowed. However,

```
(prism all) assign arr(2:7:3) = 273
```

is not allowed since `:3` is not a unit stride.

- Pointers to arrays may be printed incorrectly. [4187233]

The Prism environment prints pointers to arrays incorrectly when the pointers point to strides of other Fortran arrays.

Fortran 90 allows you to make a pointer to an array point to a stride of an original array. Thus,

```
real, dimension(10), target :: r_arr  
real, dimension(:), pointer :: p_arr  
....  
p_arr => r_arr(1:10:2)
```

is legal in Fortran 90.

The result of the previous assignment is that `p_arr` points to an array corresponding to the odd positions in `r_arr`.

Assuming `r_arr` is 1,2,3,4,5,6,7,8,9,10, then `p_arr` is 1,3,5,7,9. The Prism environment cannot operate with such strides.

```
(prism) print p_arr
```

produces 1,2,3,4,5 instead of 1,3,5,7,9.

```
(prism) assign p_arr = 0
```

changes `r_arr` to 0,0,0,0,0,6,7,8,9,10 instead of 0,2,0,4,0,6,0,8,0,10.

- Fortran 90 complex parameters printed incorrectly [4177970]

Although the Prism environment prints complex variables correctly, it does not print complex parameters correctly. For example, consider the complex parameter `P8`, declared as follows:

```
complex*8, parameter :: P8 = (1.0, -1.0)
```

The Prism command to print `P8` results in printing the wrong value:

```
(prism) p p8  
p8 = (0.000000,0.000000)
```

The result should be `p8 = (1.0, -1.0)`, as declared.

- Events involving variable length arrays may assume incorrect array dimensions. [4311934]

The Prism environment may treat events involving variable length arrays incorrectly, assuming the wrong dimensions for the arrays.