Solaris 7 Software Developer Supplement

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303–4900
U.S.A.

Part No: 806-1651–10
November 1999

Contents

Preface

The *Solaris 7 Software Developer Supplement* describes how to use new developer features in the Solaris™ 7 11/99, 8/99, 5/99, and 3/99 software releases.

Related Books

This document describes new or changed functionality in Solaris update releases. The information here supplements or supersedes information in the previous releases of Solaris 7 documentation sets.

Ordering Sun Documents

The Sun Software Shop stocks select manuals from Sun Microsystems, Inc. You can purchase individual printed manuals and AnswerBook2™ CDs.

For a list of documents and how to order them, visit the Software Shop at `http://www.sun.com/software/shop/`.

Accessing Sun Documentation Online

The docs.sun.com[SM] web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

What Typographic Conventions Mean

The following table describes the typographic changes used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls –a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

CHAPTER 1

What's New at a Glance

This chapter highlights new features added to the Solaris™ 7 operating environment in the 11/99, 8/99, 5/99, and 3/99 releases.

TABLE 1–1 Solaris 7 Features

Feature	First Released in . . .	Description
Installation (These features are described in the *Solaris 7 Installation Collection*.)		
AnswerBook2 `ab2cd` script updates	8/99	Updates to the `ab2cd` script enable users to read AnswerBook2™ documentation directly from the Solaris Documentation CD.
DNS configuration during system identification	5/99	Prior Solaris releases only allowed the configuration of a machine as a NIS or NIS+ client. With the Solaris 7 5/99 release, systems can be configured by the system identification utilities to be DNS clients.
Product Registry	3/99	The Solaris Product Registry is a tool to manage installed software. It enables you to: ■ View a list of installed and registered software and some software attributes ■ Uninstall software ■ Browse for and launch an installer
Patch Analyzer	3/99	The Patch Analyzer performs an analysis on your system to determine which (if any) patches will be removed or downgraded by upgrading from a Solaris 7 operating environment to the Solaris 7 11/99 operating environment. This analysis can be performed either at upgrade time or while the system is in production mode.

TABLE 1–1 Solaris 7 Features *(continued)*

Feature	First Released in . . .	Description
(Intel Platform Edition only) Added support for PAE mode	3/99	With the release of Pentium Pro, Intel introduced a mode called Physical Address Extension (PAE) on its advanced processors. PAE mode enables you to run multiple instances of databases and memory-intensive applications, and to support large numbers of online users on your machine. This feature is described in the *Solaris 7 11/99 Release Notes*.
System and Network Administration (These features are described in the *Solaris 7 System Administration Supplement*.)		
Universal Disk Format (UDF) file system	11/99	The UDF file system is the industry-standard format for storing information on optical media called digital versatile disc or digital video disc (DVD) that is both readable and re-writable.
Digital versatile disk or digital video disk	11/99	This release supports digital versatile disc or digital video disc (DVD) that is both readable and re-writable.
(Intel Platform Edition only) PCI hot-plugging with `cfgadm` command	11/99	The `cfgadm` command is updated to provide PCI hot-plugging for supported PCI controllers on x86 systems.
Improved device configuration (`devfsadm`)	11/99	The `devfsadm` command manages the special device files in the `/dev` and `/devices` directories and is updated to detect device configuration changes by any reconfiguration event.
(SPARC Platform Edition only) Inter-Domain Network (IDN)	11/99	Inter-Domain Network (IDN) supports high-speed networking between Dynamic System Domains (DSDs). This enables DSDs to communicate with each other using standard networking interfaces, such as TCP/IP, without the use of cabling or special hardware. See your hardware manufacturer's documentation for information about IDN and to see if IDN supports your server.
`coreadm`	8/99	The `coreadm` command provides improved core file management for Solaris system administrators.
Enhanced SCSI hot-plugging with the `cfgadm`	8/99	The `cfgadm` command now supports SCSI hot-plugging for supported SCSI hardware.
Migrating to BIND 8.1	8/99	This section updates Bind 8.1 in the *Solaris Naming Setup and Configuration Guide* and the *Solaris Naming Administration Guide*.
AnswerBook2 server software updates	8/99	Updates in AnswerBook2 version 1.4.1 server software improve navigation and performance.

TABLE 1–1 Solaris 7 Features *(continued)*

Feature	First Released in . . .	Description
Remote console	5/99	New system console features, including the `consadm` command, enable system administrators and service providers to remotely troubleshoot systems by dialing in to a system using a modem connected to one of the serial ports and to redirect console messages to the auxiliary device.
(SPARC™ Platform Edition only) Dynamic reconfiguration	5/99	Dynamic reconfiguration allows the service provider to add or remove and replace hot-pluggable system boards in a running system, eliminating the time lost in rebooting. For more information, see your hardware manufacturer's documentation.
Deferred access time updates on UFS file systems	3/99	New mount options, `dfratime` and `nodfratime`, enable and disable deferred access time updates on UFS file systems.
Improved system boot and error message format	3/99	The system boot and error message format is improved by adding a numeric identifier, module name, and time stamp to messages generated by the `syslog` logging facility.
(SPARC Platform Edition only) Improved handling of `cron` jobs on suspended systems	3/99	When you use Power Management™ software to suspend and resume a system, there is now a check on whether `cron` jobs were scheduled during the time the system was suspended. If there were, the first instance of a `cron` job is performed when the system resumes. (This improvement was available in the Solaris 7 release and first documented in the Solaris 7 3/99 release.)
Developer		
(SPARC Platform Edition) Dynamic reconfiguration support for SCSI HBA drivers	11/99	In this release, dynamic reconfiguration (DR) support has been updated for SCSI devices. See your hardware manufacturer's documentation to see if your server supports DR.
PCI hot-plugging	11/99	PCI hot-plugging is updated to include information on writing device drivers for x86 servers.
8–bit visual support	8/99	The 8-bit visual shared library enables device drivers with only 24-bit hardware to display 8-bit visual applications.
Solaris 7 64-bit Developer's Guide updates	3/99	This is a list of updates in the *Solaris 7 64-bit Developer's Guide*.

TABLE 1–1 Solaris 7 Features *(continued)*

Feature	First Released in . . .	Description
Updated DDI interfaces for cluster-aware device drivers	3/99	This overview introduces the concept of device classes and the necessary interface modifications and additions for device driver writers.
Java Development Kit		
JDK 1.1.7_08	11/99	The JDK™ 1.1.7_08 includes: ■ Enhanced Eurocurrency support ■ High-performance Just In Time compiler A Java™ Runtime Environment (JRE) corresponding to the latest version of the JDK is available for download from the Web. `http://www.sun.com/solaris/java`
Java 2 SDK 1.2.1_03	11/99	The Java 2 SDK 1.2.1_03 is the first Solaris release of Java technology based on Java 2. It includes: ■ Substantially increased scalability and performance ■ Enhanced memory management system ■ High-performing, scalable Java Virtual Machine (JVM) ■ Fast Java thread synchronization ■ Enhanced diagnostics with the heap inspection tool, which finds memory leaks in programs ■ Optimizing Just In Time compiler
Desktop User (These features are described in the *Solaris 7 User Supplement*.)		
Personal Digital Assistant (PDA) synchronization	11/99	PDA synchronization enables data to be synchronized from Sun applications, such as Calendar Manager, with data in a similar application on your PDA. Also, applications and databases can be installed from your workstation or server to your PDA.

TABLE 1–1 Solaris 7 Features *(continued)*

Feature	First Released in . . .	Description
Netscape™ application launcher	11/99	The Desktop-Apps folder window has four new icons: ■ Netscape ■ Netscape Composer ■ Netscape Mail ■ Netscape News
Xserver 11R6.4	11/99	The Xserver 11R6.4 provides the following new features: ■ XPrint ■ Xkeyboard ■ Display Power Management Signalling ■ Xinerama ■ Color Utilization Policy ■ WebEnabledX New man pages are available for each of these features.

CHAPTER 2

Software Developer

This chapter describes the following software development information:

Note - For the most up-to-date man pages, use the `man` command. The Solaris 7 11/99 man pages include new feature information not found in the *Solaris 7 Reference Manual Collection*.

SPARC: DR Support for SCSI HBA Drivers

This functionality is new in the Solaris 7 11/99 software release.

In this release, dynamic reconfiguration (DR) support has been updated for SCSI devices. SCSI HBA drivers no longer require a `cb_ops` structure to support dynamic reconfiguration, as was previously described in, "Converting Device Drivers to Support Hot-plugging" in *Writing Device Drivers*.

New Default SCSI HBA Driver Entry Points

To support the minimal set of hot-plugging operations, drivers might need to implement support for bus quiesce, bus unquiesce, and reset. The `scsi_hba_tran(9S)` structure has been extended to support these new operations. If quiesce/unquiesce/reset is not required by hardware, no driver changes are needed.

The following new fields have been added to the `scsi_hba_tran` structure:

```
int (*tran_quiesce)(dev_info_t *hba_dip);
int (*tran_unquiesce)(dev_info_t *hba_dip);
int (*tran_bus_reset)(dev_info_t *hba_dip, int level);
```

The new driver entry points are introduced in the following sections.

`tran_quiesce()` and `tran_unquiesce()`

Quiesce and unquiesce a SCSI bus.

```
#include <sys/scsi/scsi.h>

int prefixtran_quiesce(dev_info_t *hba_dip);

int prefixtran_unquiesce(dev_info_t *hba_dip);
```

`tran_quiesce(9E)` and `tran_unquiesce(9E)` are required to be implemented by an HBA driver to support dynamic reconfiguration (DR) of SCSI devices on buses that were not designed to support hot-plugging.

The `tran_quiesce()` and `tran_unquiesce()` vectors in the `scsi_hba_tran(9S)` structure should be initialized during the HBA driver's `attach(9E)` to point to HBA entry points so they are called when a user initiates quiesce and unquiesce operations.

`tran_quiesce(9E)` is called by the SCSA framework to stop all activity on a SCSI bus prior to and during the reconfiguration of devices attached to the SCSI bus. `tran_unquiesce(9E)` is called by the SCSA framework to resume activity on the SCSI bus after the reconfiguration operation has been completed.

HBA drivers are required to handle `tran_quiesce(9E)` by waiting for all outstanding commands to complete before returning success. After the HBA has quiesced the bus, it must queue any new I/O requests from target drivers until the SCSA framework calls the corresponding `tran_unquiesce(9E)` entry point.

HBA drivers handle calls to `tran_unquiesce(9E)` by starting any target driver I/O requests that were queued by the HBA during the time the bus was quiesced.

`tran_bus_reset()`

`tran_bus_reset(9E)` must reset the SCSI bus without resetting targets.

```
#include <sys/scsi/scsi.h>

int prefixtran_bus_reset(dev_info_t *hba_dip, int level);
```

Where `level` is:

`RESET_BUS` reset the SCSI bus only, not the targets

The `tran_bus_reset( )` vector in the `scsi_hba_tran(9S)` structure should be initialized during the HBA driver's `attach(9E)` to point to an HBA entry point to be called when a user initiates a bus reset.

Implementation is hardware specific. If it is not possible to reset the SCSI bus without affecting the targets, the HBA driver should fail `RESET_BUS` or not initialize this vector.

For more information, see "Converting Device Drivers to Support Hot-plugging" in the book *Writing Device Drivers*.

Special Issues with PCI Hot-plugging Drivers

This functionality is new in the Solaris 7 11/99 software release.

Some PCI device drivers for the Intel platform are written with the assumption that the values in the registers are valid after the `POWER ON` cycle. However, you cannot assume that the `BIOS` has correctly initialized the hardware.

For more information on hot-plugging, see "Converting Device Drivers to Support Hot-plugging" in the book *Writing Device Drivers*.

8-bit Visual Support

This functionality is new in the Solaris 7 8/99 software release.

The 8–bit visual shared library provides a set of translation functions, enabling 8-bit visual applications to run on hardware that only provides support for 24-bit visual depth. The functions use the device driver's native 24-bit rendering function calls for applications requesting 8-bit visual support. This is done by translating 8-bit pseudocolor colormap pixel data into 24-bit truecolor colormap pixel data before rendering an image on the 24-bit hardware visual supported platform.

64–bit Developer's Guide Updates

This information was new in the Solaris 7 3/99 software release.

The following updates are in the *Solaris 7 64-bit Developer's Guide*.

- SPARC Compliance Definition, Version 2.4 is now available from SPARC International at `http://www.sparc.com`.
- Information regarding the 64–bit version of `ls` has been added to Appendix B.
- Compiler command references have been updated.

  Since the release of the Solaris 7 operating environment, the Sun Workshop™ Compilers C and its accompanying user's guide have been updated. All references to the `lint` program and the C compiler have been updated. For additional information, see `cc(1)` and `lint(1)`.

DDI Interfaces for Cluster-Aware Drivers

This feature was new in the Solaris 7 3/99 software release.

The device node types supported by the Solaris operating environment can be divided into two categories: physical and pseudo devices. This categorization is important when the device nodes are created and used by Sun™ Cluster.

The concept of device classes and the necessary interface modifications and additions are introduced in the 3/99 release of the Solaris operating environment so that device driver writers can adopt the new interfaces for use with future versions of Sun Cluster. The device classes will not have an impact on Solaris operation because they are ignored by the base kernel without Sun Cluster software installed.

For more information regarding device drivers, see *Writing Device Drivers*.

Device Classification

Sun Cluster introduces four new device classes. These new classifications are based on the extended behavior of the devices in a Sun Cluster environment.

Enumerated Devices	`ENUMERATED_DEV`
Node Specific Devices	`NODESPECIFIC_DEV`

Global Devices	GLOBAL_DEV
Node Bound Devices	NODEBOUND_DEV

The ddi_create_minor_node(9F) routine has been enhanced to add the capability of reporting the additional device classification of the device minor nodes created by the device driver. The device categories are described in the following sections.

Enumerated Devices

Enumerated Devices are physical devices with a one-to-one correspondence between a particular device node and a host where that device node is present. Examples of this category include various disk and tape devices, such as /dev/dsk/c0t0d0s0 and /dev/rmt/0l. Nearly all physical devices belong to this category. This is the default category for all non-pseudo devices.

Node Specific Devices

Node Specific Devices include devices that report particular information about the host where the device node is opened. An example of such a device is the /dev/kmem device. Opening this device gives access to host-specific information on the local host. Administrative pseudo device nodes used in configuring or gathering information about a particular device driver also fit this category. The Sun Cluster software ensures the creation of two user device nodes for each instance of a kernel device node in the cluster, so that the intended device node can be accessed both locally and remotely.

Global Devices

Global Devices are node invariant pseudo devices such as /dev/ip. In principle, the open instance of a device, such as ip or tcp, does not depend on which host, in the cluster, the open occurs. A single copy of each device is in the kernel. All device I/O requests for this device class are performed locally and the device node can be accessed by a remote host within the cluster. This is the default behavior for all pseudo devices in the system.

Node Bound Devices

A node bound device is a pseudo device that maintains a cluster-wide state. This device should, in principle, be opened on one node only. Devices such as /dev/ticotsord belong to this class. Highly available devices with automatic fail-over also belong to this class. Only one pseudo node is present but all opens are

directed to the same node, with the exception of HA devices, where the hosting node might change transparently to the device user.

Minor Number Space Management

dev_t consists of a major and a minor number space. Major number space is managed by Solaris and the minor number space is managed by the device driver space. With Sun Cluster, the minor number behaves differently within the user space and the kernel space.

Cluster Wide dev_t

For historical reasons each device node, in addition to its path, is identified by an integral type dev_t. The dev_t is a part of the system interface expected by programmers and system administrators. stat(2) system calls and backup utilities deal directly with dev_ts. dev_t is also a programming interface for device driver writers.

Sun Cluster preserves the assumption that two equal dev_ts point to the same device regardless of the host where the process is executed. This model satisfies the expectations of programs that depend on this feature to establish the equivalence of two devices. Sun Cluster introduces a dual view of minor numbers and the necessary interfaces to implement this dual view. In kernel dev_ts correspond to the major number of the driver in addition to the minor number that the driver has created using ddi_create_minor_node(9F). External minor numbers (viewed from the user space) are managed and assigned unique cluster-wide numbers by the device configuration manager in Sun Cluster.

This dual numbering scheme has one unfortunate side effect, namely that a particular minor number created in the kernel can result in creation of a different minor number in the user space. This discrepancy might be unexpected by user space programs that expect to be able to ascertain some device characteristics from the minor number pattern.

An example of the discrepancy is the use of minor number bit patterns in specifying the particular slice of a disk or the density of a tape device. This class of problems is primarily alleviated by the use of globally unique instance numbers. By encoding the instance number of a device in the minor, the driver can guarantee the creation of cluster-wide unique dev_t values; this avoids minor numbers that do not have the same value between the kernel and the user space.

All dev_t values that are passed in through the standard Solaris entry points such as open, close and ioctl, encode the kernel minor number. The getminor(9F) interface can be used to extract this minor number. However, if the dev_t value is passed as a part of the ioctl data from the user space, the dev_t value has the minor number from the user space encoded. A new DDI interface,

ddi_getiminor(9F), has been introduced to ensure that the driver can map between internal and external minor numbers.

Device Interfaces

The following interface sets up a driver and prepares it for use:

```
int ddi_create_minor_node(dev_info_t *dip, char *name,
   int spec_type, int minor_num, char *node_type, int flag);
```

ddi_create_minor_node(9F) advertises a minor device node, which will eventually appear in the /devices directory and refer to the device specified by dip. If the device is a clone device, then flag is set to CLONE_DEV. If it is not a clone device, then flag is set to 0. For device drivers intended for use in a clustered environment, flag must specify the device node class of GLOBAL_DEV, NODEBOUND_DEV, NODESPECIFIC_DEV, or ENUMERATE_DEV.

The following new interface is used to translate between user-visible device numbers and in kernel device numbers:

```
minor_t ddi_getiminor(dev_t dev);
```

ddi_getiminor(9F) extracts the minor number as a device number.

CHAPTER 3

Java for Developers

This chapter describes Java enhancements for developers.

Note - The 8/99 update release included the JDK 1.1.7_07 release.

Note - For the most up-to-date man pages, use the `man` command. The Solaris 7 11/99 man pages include new feature information not found in the *Solaris 7 Reference Manual Collection*.

Java 2 SDK 1.2.1_03

These enhancements are new in the Solaris 7 11/99 software release. For more information, see *Java 2 SDK Developer's Guide for Solaris*.

Heap Inspection Tool

This diagnostic tool for interactively killed programs is accessible from the SIGQUIT handler menu. It can be used to find memory leaks in your programs. A memory leak occurs when a program inadvertently retains objects, preventing the garbage collector from reclaiming the memory. Heap inspection presents a per-class breakdown of the

objects in the heap, sorted by total amount of memory consumed. You can then examine reference chains to selected objects to see what is keeping them alive.

Double-Word Alignment to 8-Byte Boundaries in the Heap

Double-word (longs and doubles) values are now aligned to 8-byte boundaries in the heap. This improves the performance of both native code and JIT-compiled Java code while ensuring correctness of volatile double-word values on SPARC systems. However, if your application allocates and retains many small objects, you may need to increase your heap size(s) slightly, since these objects will be allocated in multiples of 8 bytes, increasing your memory usage.

Optimizing JIT Compiler

The Java 2 SDK includes an optimizing JIT compiler which improves performance without sacrificing application start-up time. Specifically, the JIT compiler has improved in its ability to identify optimization opportunities, translating frequently invoked methods and methods with loops into highly efficient native code.

Enhanced Memory Management System

The Java Virtual Memory (JVM) also includes a highly optimized memory system, making memory allocation and garbage collection more efficient. It is a non-conservative, fully compacting, generational memory system which uses direct pointers. This feature increases batch program performance and reduces disruptive garbage collection pauses in interactive programs.

Faster Thread Synchronization

The JVM has significantly improved implementations of the Java platform's synchronization primitives. These implementations make concurrent programs more efficient and decrease the impact of the synchronization primitives on single-threaded application performance.

Java Development Kit (JDK) 1.1.7_08

This feature is new in the Solaris 7 11/99 software release.

Enhanced Euro Currency Support

The JDK 1.1.7_07 release supports the new European Union currency, the euro, including these changes:

- Unicode support is upgraded from version 2.0.14 to version 2.1.2.
- Character encodings have changed and new encodings have been added.
- New locales have been added.

In order for the JDK software support for the euro character to function properly, you may need to install euro-support patches for your operating system. Euro-support patches are required if you are using versions of Solaris software prior to the Solaris 7 operating environment. Check with your Solaris support provider or visit the SunSolve[SM] website at `http://sunsolve.sun.com` for information on the availability of the Solaris euro-support patches.

The addition of euro support will not affect existing code, except for character conversion code relying on changed encodings. No APIs have been changed.

For information on the euro, see the euro web site at `http://europa.eu.int/euro/`.

Java Runtime Environment (JRE) Available

A JRE corresponding to the latest version of the JDK is available for download at `http://www.sun.com/solaris/jre`.

Refer to the document, *Solaris Java Runtime Environment (JRE) Configuration Guide* at `http://www.sun.com/solaris/jre/download.1.1.7/en/jre_config.txt`.

This document contains the following information:

- JRE overview
- How to bundle and run the JRE

The bugs in these release notes apply to the JRE, except for those bugs that apply to components not included in the JRE.

High-Performance JIT Compiler

JDK 1.1.6_03 introduced the High-Performance JIT compiler, an optimized version that enables significant performance gains over the JIT compiler bundled with the Solaris 2.6 environment.