Solaris 10 What's New

Contents

Preface

Solaris 10 What's New summarizes all features in the Solaris™ 10 release that are new or have been enhanced since the Solaris 9 Operating System (Solaris OS) was originally distributed in May 2002.

Note – This Solaris release supports systems that use the SPARC® and x86 families of processor architectures: UltraSPARC®, SPARC64, AMD64, Pentium, and Xeon EM64T. The supported systems appear in the *Solaris 10 Hardware Compatibility List* at `http://www.sun.com/bigadmin/hcl`. This document cites any implementation differences between the platform types.

In this document the term "x86" refers to 64-bit and 32-bit systems manufactured using processors compatible with the AMD64 or Intel Xeon/Pentium product families. For supported systems, see the *Solaris 10 Hardware Compatibility List*.

In the Solaris 10 OS, Sun Microsystems has developed a new architecture for building and deploying systems and services that are capable of Predictive Self-Healing. See "Predictive Self-Healing" on page 16. Also, changes in the installation of the Solaris OS provide a simplified and unified installation process. See "Solaris Installation Changes Including Installation Unification" on page 43.

The Solaris Zones partitioning technology is a key feature that has been added to the Solaris 10 OS. Zones are used to virtualize operating system services and provide an isolated and secure environment for running applications. See "Solaris Zones Software Partitioning Technology" on page 35. Additional key features in the Solaris 10 OS include "Process Rights Management" on page 52, "DTrace Dynamic Tracing Facility" on page 18, and "New Architecture for Network Stacks" on page 67. The "Java Desktop System, Release 3" on page 33 is now also available in the Solaris 10 OS.

Who Should Use This Book

This book provides introductory descriptions of the new Solaris 10 features for users, developers, and system administrators who install and use the Solaris 10 Operating System.

Optional Feature Licensing

Certain optional features and products that are described in this document might require individual licensing for use. Refer to the Software License Agreement.

Related Books

For further information about the features that are summarized in this book, refer to the following Sun documentation at `http://docs.sun.com`.

- *Application Packaging Developer's Guide*
- *Java Desktop System Release 3 Collection*
- *International Language Environments Guide*
- *Linker and Libraries Guide*
- *Programming Interfaces Guide*
- *Solaris 10 Release Notes*
- *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*
- *Solaris 10 Installation Guide: Basic Installations*
- *Solaris 10 Installation Guide: Network-Based Installations*
- *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*
- *Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning*
- *Solaris 64-bit Developer's Guide*
- *Solaris Dynamic Tracing Guide*
- *Solaris Security for Developers Guide*
- *Solaris Smartcard Administration Guide*
- *Solaris System Management Agent Administration Guide*
- *Solaris System Management Agent Developer's Guide*
- *Solaris Tunable Parameters Reference Manual*
- *Solaris Volume Manager Administration Guide*
- *STREAMS Programming Guide*
- *System Administration Guide: Advanced Administration*
- *System Administration Guide: Basic Administration*
- *System Administration Guide: Devices and File Systems*
- *System Administration Guide: IP Services*

- *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
- *System Administration Guide: Network Services*
- *System Administration Guide: Security Services*
- *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*
- *Writing Device Drivers*
- *x86 Assembly Language Reference Manual*

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

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`.

Ordering Sun Documentation

Sun Microsystems offers select product documentation in print. For a list of documents and how to order them, see "Buy printed documentation" at `http://docs.sun.com`.

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
AaBbCc123	What you type, contrasted with onscreen computer output	`machine_name%` **su** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms or terms to be emphasized	Read Chapter 6 in the *User's Guide*. Do *not* save the file.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

What's New in Solaris 10

This book summarizes all features in the Solaris 10 Operating System (Solaris OS) that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002.

For a summary of features that were introduced in the Solaris 9, Solaris 8, or Solaris 7 releases, see *What's New in the Solaris 9 Operating Environment* at `http://docs.sun.com`. This book summarizes Solaris 9 features, and fully describes Solaris 8 and Solaris 7 features in the appendix.

Key Features in Solaris 10 Software

In the Solaris 10 OS, Sun Microsystems has developed a new architecture for building and deploying systems and services that are capable of Predictive Self-Healing. Self-healing technology enables Sun systems and services to maximize availability. See "Predictive Self-Healing" on page 16. The Predictive Self-Healing feature includes significant changes to the booting and service administration processes. See "Solaris Service Manager" on page 16. Also in the Solaris 10 OS, changes in the installation of the operating system provide a simplified and unified installation process. See "Solaris Installation Changes Including Installation Unification" on page 43.

The Solaris 10 OS introduces Solaris Zones software partitioning technology. Solaris Zones is a component of the Solaris Containers environment. Zones are used to virtualize operating system services and provide an isolated and secure environment for running applications. See "Solaris Zones Software Partitioning Technology" on page 35.

Other key features in the Solaris 10 software include the DTrace dynamic tracing facility, process rights management, and a new architecture for network stacks. DTrace is a comprehensive dynamic tracing facility that gives Solaris users, administrators, and developers a new level of observability into the kernel and user processes. For further information, see "DTrace Dynamic Tracing Facility" on page 18. In the Solaris software, processes that previously required superuser capabilities now require process rights. Process rights management uses privileges to restrict processes to only those rights that are required to perform the task. See "Process Rights Management"

on page 52 for further information. Also of particular importance, the networking stack for TCP connections has been rearchitected to deliver extremely high performance while improving the scalability. For further information, see "New Architecture for Network Stacks" on page 67.

Java™ 2 Platform, Standard Edition 5 is another key technology. See "Java 2 Platform, Standard Edition 5" on page 40 for descriptions of the enhancements that are available in this platform. Also of particular significance, the Solaris 10 software introduces support for the 64–bit computing capabilities of the AMD Opteron processor. For details, see "64-bit Support on x86 Systems" on page 19. Finally, the Solaris 10 software introduces the Java Desktop System, Release 3. This desktop system combines open source software with Sun innovation. For further information, see "Java Desktop System, Release 3" on page 33.

These key features and many other new features in the Solaris 10 software are summarized in this manual. Most Solaris 10 features were introduced through the Software Express program. To view a list of all Solaris 10 features by their original Solaris Express release date, see "Features by Release Date" on page 123.

System Administration Tools

This section describes all system administration features in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002. DTrace and Predictive Self-Healing are features of particular importance.

Predictive Self-Healing

This feature was introduced in the Solaris Express 6/04 release. The Solaris Express 10/04 release and the Solaris 10 release provided important enhancements.

Sun Microsystems has developed a new architecture for building and deploying systems and services capable of Predictive Self-Healing. Self-healing technology enables Sun systems and services to maximize availability when software and hardware faults occur. In addition, the self-healing technology facilitates a simpler and more effective end-to-end experience for system administrators and service providers, thereby reducing costs. The first major set of new features to result from this initiative is available in the Solaris 10 OS. The Solaris 10 software includes components that facilitate self-healing for CPU, memory, I/O bus nexus components, and system services.

Specific details about the components of this new architecture are covered in the following descriptions for the Solaris Service Manager and the Solaris Fault Manager.

Solaris Service Manager

Introduced in the Solaris Express 10/04 release and enhanced in the Solaris 10 release, the Solaris Service Manager provides an infrastructure that augments the traditional UNIX® startup scripts, init run levels, and configuration files. This infrastructure provides the following functions:

- Automatically restarts failed services in dependency order, whether the services failed as the result of administrator error, a software bug, or an uncorrectable hardware error.

- Makes services objects that can be viewed, with the new `svcs` command, and managed, with `svcadm` and `svccfg` commands. You can also view the relationships between services and processes by using `svcs -p`, for both SMF services and legacy `init.d` scripts.

- Makes it easy to back up, restore, and undo changes to services by taking automatic snapshots of service configurations.

- Makes it easy to debug. You can ask questions about services and receive an explanation of why a service isn't running by using `svcs -x`. Also, this process is eased by individual and persistent log files for each service.

- Enhances the ability of administrators to securely delegate tasks to nonroot users, including the ability to modify properties and start, stop, or restart services on the system.

- Boots faster on large systems by starting services in parallel according to the dependencies of the services. The opposite process occurs during shutdown.

- Enables you to customize the boot console output either to be as quiet as possible, which is the default, or to be verbose by using `boot -m verbose`.

- Preserves compatibility with existing administrative practices wherever possible. For example, most customer and ISV-supplied rc scripts still work as usual.

- Enables you to configure your system services in one of two modes, both represented as `smf`(5) profiles. The "generic_open.xml" profile enables all the traditional Internet services that were previously enabled by default in the Solaris OS. The "generic_limited_net.xml" profile disables a large number of services that are frequently disabled during the process of hardening a system. However, this profile is not a replacement for the Solaris Security Toolkit (JASS) tool. See the individual profiles for details.

See Chapter 9, "Managing Services (Overview)," in the *System Administration Guide: Basic Administration* for more information about this infrastructure. An overview of the infrastructure can be found in the `smf`(5) man page.

Solaris Fault Manager

Predictive Self-Healing systems include a simplified administration model. Traditional error messages are replaced by telemetry events that are consumed by software components. The software components automatically diagnose the underlying fault or defect and initiate self-healing activities. Examples of self-healing activities include administrator messaging, isolation or deactivation of faulty components, and guided repair. A new software component is called Fault Manager, `fmd`(1M). The Fault Manager manages the telemetry, log files, and components. The new `fmadm`(1M), `fmdump`(1M), and `fmstat`(1M) tools are also available in the Solaris 10 OS to interact with the Fault Manager and new log files.

When appropriate, the Fault Manager sends a message to the syslogd(1M) service to notify an administrator that a problem has been detected. The message directs administrators to a knowledge article on Sun's new message Web site, http://www.sun.com/msg/, which explains more about the problem impact and appropriate responses and repair actions.

The Solaris Express 6/04 release introduced self-healing components for automated diagnosis and recovery for UltraSPARC-III and UltraSPARC-IV CPU and memory systems. This release also provided enhanced resilience and telemetry for PCI-based I/O.

DTrace Dynamic Tracing Facility

DTrace is a feature of key importance in the Solaris 10 OS. DTrace was introduced in the Solaris Express 11/03 release and was enhanced in the Solaris Express 5/04 release and the Solaris Express 11/04 release.

DTrace is a comprehensive dynamic tracing facility that gives Solaris users, administrators, and developers a new level of observability into the kernel and user processes.

DTrace helps you understand your system by permitting you to dynamically instrument the operating system kernel and user processes to record data that you specify at locations of interest, called *probes*. Probes are like small programmable sensors that are scattered all over your Solaris system. More than 30,000 probes are included in this release. Each probe can be associated with custom programs that are written in the new D programming language. This association enables you to access system data by using ANSI C types and expressions and easily capture stack traces, record timestamps, build histograms, and more.

All of DTrace's instrumentation is entirely dynamic and available for use on your production system. When DTrace is off, you have no performance cost. The performance impact of any tracing is limited to only those probes and actions that you enable. Moreover, DTrace is safe: you can't damage the running system because DTrace has security, complete safety, and error checking at the core of its design. These features enable you to use DTrace with confidence on your running system whenever you need the facility to help investigate a problem.

The *Solaris Dynamic Tracing Guide* and a set of man pages, including dtrace(1M), are provided to help you learn DTrace. The guide includes a complete feature reference and examples to help you get started. You can also join the DTrace forum on BigAdmin (see http://www.sun.com/bigadmin/content/dtrace) to learn more about DTrace, download example programs, and ask questions.

In the Solaris Express 5/04 release, user-process tracing that uses the pid provider was made available on x86. This feature has been available on SPARC platforms since the introduction of DTrace in the Solaris Express 11/03 release.

The `pid` provider enables you to trace any instruction in any process either at the level of any function call's entry and return, or at any offset into any function. For complete details, see Chapter 27, "pid Provider," and Chapter 32, "User Process Tracing," in the *Solaris Dynamic Tracing Guide*.

In the Solaris Express 11/04 release, the `plockstat`(1M) utility enables you to observe user-level synchronization primitives, such as `lockstat`(1M) in the kernel. The DTrace `plockstat` provider is the underlying instrumentation methodology for `plockstat`(1M). Dtrace `plockstat` can be used to augment the data recorded by the `plockstat` utility. See the `plockstat`(1M) man page for further information.

64-bit Support on x86 Systems

This feature was introduced in the Solaris Express 11/04 release.

The Solaris 10 release contains support for the 64-bit computing capabilities of AMD64 and EM64T platforms. This support includes the ability to run both 32-bit and 64-bit applications in the 64-bit computing environment.

For details about supported hardware configurations, see `http://www.sun.com/bigadmin/hcl`.

Note – The Solaris 10 software already supports these platforms for 32-bit computing.

For more information about Solaris for 64-bit support on x86 systems, see the following references in the Solaris 10 documentation:

- Appendix C, "Making a Device Driver 64-bit Ready" in *Writing Device Drivers*
- Chapter 8, "Shutting Down and Booting a System (Overview)" in *System Administration Guide: Basic Administration*
- *Solaris 64-bit Developer's Guide*

Kernel Selection for x86 Systems

This feature was introduced in the Solaris Express 11/04 release.

The Solaris 10 OS has the ability to automatically detect whether your system is 64-bit capable and then boot the appropriate kernel.

Following a new installation of the Solaris 10 software, the boot program automatically loads the 64-bit kernel if it detects your system is 64-bit capable. Otherwise, the program loads the 32-bit kernel.

Following an upgrade installation of the Solaris 10 OS on a system that is configured to load the default 32-bit kernel, the system now automatically determines whether to load the 32-bit or 64-bit kernel. If the system was configured to load a non-default kernel, the system continues to load that non-default kernel. Procedures for customizing a system to load a specific kernel are outlined in Chapter 8, "Shutting Down and Booting a System (Overview)" in *System Administration Guide: Basic Administration*.

Further documentation about the kernel selection procedure is provided in the Solaris 10 documentation at `http://docs.sun.com`.

Sun Java Web Console

This feature was introduced in the Solaris Express 10/04 release.

The Sun Java Web Console provides a common location for users to access web-based management applications. Users reach the console by logging in through an HTTPS port, using one of several supported web browsers. The single entry point that is provided by the console eliminates the need to learn URLs for multiple applications. This entry point provides authentication and authorization for all applications that are registered with the console.

All console-based applications conform to the same user interface guidelines. This change enhances ease of use by enabling users to transfer their knowledge of one application to another.

The Java Web Console also provides auditing and logging services for all registered applications.

For more information about the Java Web Console, see the *System Administration Guide: Basic Administration*.

Multi-Owner Disk Set Support With Solaris Volume Manager for Sun Cluster

This feature was introduced in the Solaris Express 10/04 release and in the Solaris 9 9/04 release.

Solaris Volume Manager for Sun Cluster introduces volume management features that work specifically with Sun Cluster and with applications such as Oracle Real Application Clusters. Solaris Volume Manager for Sun Cluster enables you to create and manage storage that is grouped into multi-owner disk sets. A multi-owner disk set enables multiple nodes to share ownership of a disk set. Multi-owner disk sets provide scalability because an instance of the application runs on each node in the cluster. Because each instance of the application directly accesses the shared storage, multi-owner disk sets also enhance the performance of the application.

For more information, see the *Solaris Volume Manager Administration Guide*. For information about Sun Cluster, see *Sun Cluster Software Installation Guide for Solaris OS* and *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.

Expanded Disk Set Support in the Solaris Volume Manager

This feature was introduced in the Solaris Express 1/04 release and in the Solaris 9 9/04 release.

The Solaris Volume Manager has expanded the functionality of device IDs. In addition to tracking movement in local sets, the Solaris Volume Manager now also automatically tracks disk movement within named disk sets. This enhancement ensures that Solaris Volume Manager volumes remain intact even if disks are rearranged. This device ID support also enables two hosts to share access nonconcurrently to disk sets, even with dissimilar views of accessible raw storage.

The Solaris Volume Manager has also been enhanced with the metaimport command. This command uses the expanded device ID support to enable you to import disk sets, even disk sets that were created on different systems.

See the metaimport(1M) man page for further information.

Importing Remotely Replicated Disk Sets With Solaris Volume Manager

This feature was introduced in the Solaris Express 10/04 release.

Solaris Volume Manager has expanded the functionality of the metaimport command. In addition to importing regular disk sets from one system to another, you can now import replicated disk sets.

A replicated disk set is created by using remote replication software, such as Sun StorEdge™ Network Data Replicator (SNDR) software or Hitachi TrueCopy.

For more information, see the metaimport(1M) man page and the *Solaris Volume Manager Administration Guide*.

Device ID Changes

This description was introduced in the Solaris Express 10/04 release.

Device ID output is displayed in a new format in the Solaris Express 10/04 release. Previously, the device ID was displayed as a hexadecimal value. The new format displays the device ID as an ASCII string. Solaris Volume Manager might display the device ID output in the new or old format, depending on when the device id information was added to the state database replica.

For more information, see the *Solaris Volume Manager Administration Guide*.

Multiterabyte Volume Support in Solaris Volume Manager

Multiterabyte volume support is available only for systems that run a 64-bit kernel. This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

Solaris Volume Manager has been enhanced with multiterabyte volume support. With this support, the Solaris Volume Manager can create, manage, and delete large (>1Tbyte) RAID-0 (stripe), RAID-1 (mirror), RAID-5, and soft-partition volumes. In addition, this large volume support enables Solaris Volume Manager to construct volumes on large or EFI-labeled logical unit numbers (LUNs).

Solaris Volume Manager large volume support is not available for systems running a 32-bit Solaris kernel.

See also "Multiterabyte Disk Support With EFI Disk Labels" on page 106.

For further information, see the *Solaris Volume Manager Administration Guide*.

Reconfiguration Coordination Manager Support in Solaris Volume Manager

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

Reconfiguration Coordination Manager (RCM) support adds to Solaris Volume Manager the ability to respond appropriately to dynamic reconfiguration (DR) requests. This addition ensures that removal of devices under Solaris Volume Manager control is blocked with an appropriate warning. This block remains in effect until the devices are no longer in use. This warning prevents system administrators from accidentally removing active volumes from a DR-configured system.

For further information, see the *Solaris Volume Manager Administration Guide*.

Top-Down Volume Creation in Solaris Volume Manager

This feature was introduced in the Solaris Express 9/03 release and in the Solaris 9 4/04 release.

The top-down volume creation feature of Solaris Volume Manager provides a new command that enables system administrators to quickly and easily create quality of service-based Solaris Volume Manager configurations. Rather than manually partitioning disks, creating stripes, and assembling mirrors, the metassist command manages the details and provides functional logical volumes. The volumes are based on the criteria that are specified on the command line or in referenced configuration files.

For further information, see the metassist(1M) man page and the *Solaris Volume Manager Administration Guide*.

Package and Patch Tool Enhancements

This feature was introduced in the Solaris Express 11/03 release under the title "Solaris Product Registry Enhancements." This feature description was updated in the Solaris Express 8/04 and 10/04 releases.

The Solaris package and patch tools have been enhanced, providing improved performance and extended functionality.

As a part of these enhancements, the pkgchk command now provides a new option to assist you in mapping files to packages. To map files to packages, use the pkgchk -P option instead of grep *pattern* /var/sadm/install/ contents. The -P option enables you to use a partial path. Use this option with the -l option to list the information about the files that contain the partial path.

If you installed a previous Software Express release, your system might use a SQL format package database. The SQL database was created in one of the following ways.

- You performed an initial installation of a Solaris Express release prior to the Solaris Express 10/04 release.
- You upgraded to a prior Solaris Express release, and upgraded the package database manually by running pkgadm upgrade.

When you upgrade to the Solaris Express 10/04 release or subsequent releases, the SQL package database is automatically converted back to the ASCII text file format.

See the *System Administration Guide: Basic Administration* and the pkgchk(1M) man page for further information.

Enhancements for pbind and psrset Commands

These enhancements were introduced in the Solaris Express 10/04 release.

System administrators can now query and establish per-thread processor and processor set bindings. New options have been added to pbind and psrset commands. These new options enable you to query which LWPs on the system are bound to specified processor or processor sets. You can also clear all processor or processor set bindings.

See the pbind(1M) and psrset(1M) man pages for further information about these commands.

Solaris Dynamic Host Configuration Protocol (DHCP)

This feature was introduced in the Software Express pilot program.

The Solaris Dynamic Host Configuration Protocol (DHCP) service includes the DHCP Manager graphical tool and command-line tools. This Solaris release adds new options to the dhcpconfig and dhtadm commands.

With the new options for the dhcpconfig command, you can do the following:

- Enable and disable DHCP service for system starts
- Restart DHCP server
- Prompt DHCP server to read the dhcptab
- Specify DHCP server parameters

A new suboption for the dhtadm command causes the DHCP server to read the dhcptab to implement changes that were made with the dhtadm command.

The new options enable you to perform all DHCP management tasks from the command line as well as from DHCP Manager.

For more information, see the man pages for dhcpconfig(1M) and dhtadm(1M).

See also the *System Administration Guide: IP Services*.

DHCP Event Scripting

This feature was introduced in the Solaris Express 8/04 release.

DHCP is a protocol for configuring systems with network information and leased IP addresses. The DHCP client is a daemon that runs on a network host and obtains configuration parameters for the host's network interfaces from a DHCP server.

The Solaris DHCP client daemon, dhcpagent, is now able to execute administrator-defined actions. The actions can be triggered after any of the following DHCP lease events occur:

- The interface is configured for DHCP.
- The client successfully extends a lease.
- The lease expires.
- The client drops the lease to remove the interface from DHCP control.
- The client relinquishes the address.

The administrator-defined actions must be initiated by an executable script or program that you provide. You can use the program to perform any action that is appropriate for your system after one of the DHCP events.

For more information, see the dhcpagent(1M) man page.

DHCP for Logical Interfaces

This feature was introduced in the Solaris Express 8/04 release.

The DHCP client daemon, dhcpagent, can now be used with logical interfaces as well as physical interfaces. DHCP is started on logical interfaces by using the same methods that are used to start DHCP on physical interfaces.

However, the DHCP client has two limitations for the management of leases on logical interfaces:

- The DHCP client does not automatically manage default routes associated with logical interfaces.
- The DHCP client does not automatically generate a client identifier for logical interfaces.

For more information, see the dhcpagent(1M) man page.

x86: SunVTS 6.0

SunVTS™ is a comprehensive diagnostic tool that tests and validates Sun hardware by verifying the connectivity and functionality of most hardware controllers and devices on Sun platforms.

SunVTS support for x86 systems was introduced in the Solaris Express 8/04 release. The SunVTS infrastructure and a few core diagnostics are now available for x86 platforms. The current x86 support is for the 32-bit operating system only.

You must install the x86 version of the SunVTS packages in order to use SunVTS on x86 platforms. The software packages use the same names as in the SPARC environment. A new package, SUNWvtsr, is provided for both the SPARC and x86 packages for this release. The SUNWvtsr package contains the SunVTS framework configuration files in the root partition. The x86 packages are located in the `/sol_10_x86/s0/Solaris_10/ExtraValue/CoBundled/SunVTS_6.0` directory of the Solaris Software media. The SPARC packages are located in the `/sol_10_sparc/s0/Solaris_10/ExtraValue/CoBundled/SunVTS_6.0` directory of the Solaris Software media.

Refer to the SunVTS user documentation available at the following web sites for detailed information about x86 system support: `http://docs.sun.com` or `http://www.sun.com/products-n-solutions/hardware/docs/`.

Kernel Modular Debugger

Kernel Modular Debugger (KMDB) is an MDB-based kernel debugger. In the Solaris Express 8/04 release, KMDB replaced the existing kernel debugger (kadb) as the standard Solaris kernel debugger.

KMDB brings all the power and flexibility of Modular Debugger (MDB) to live kernel debugging. KMDB supports the following:

- `dcmds` – Debugger commands
- `dmods` – Debugger modules
- Access to kernel type data
- Kernel execution control
- Inspection
- Modification

KMDB can be loaded at boot time or after the system has already booted.

For more information, see the `kmdb`(1M) man page. See also the *Solaris Modular Debugger Guide* and the *System Administration Guide: Basic Administration*.

Solaris Process Accounting and Statistics Improvements

This feature was introduced in the Solaris Express 8/04 release.

In this release, changes have been made to the internals of the load averaging, `cpu usr/sys/idle`, and accounting functions. Microstate accounting has replaced the old accounting mechanism and is enabled by default. As a result, you might notice slightly different process usage and timing statistics.

Switching to microstate accounting provides substantially more accurate data about user processes and the amount of time they spend in various states. In addition, this information is used to generate more accurate load averages and statistics from the `/proc` file system.

For more information about process accounting and statistics, see the *System Administration Guide: Advanced Administration*. See also the `proc(4)` man page.

Enhancements to `ls` Command

This feature was introduced in the Solaris Express 8/04 release.

The `ls` command can now display file times to the second or to the nanosecond. Use the `-e` option for seconds and the `-E` option for nanoseconds. The following example illustrates these new options:

```
% ls -l foo
-rw-r--r--  1 jpk    staff   0 Aug  6 15:08 foo
% ls -e foo
-rw-r--r--  1 jpk    staff   0 Aug  6 15:08:28 2004 foo
% ls -E foo
-rw-r--r--  1 jpk    staff   0 2004-08-06 15:08:28.776641000 -0700 foo
```

See the `ls(1)` man page for further information about this command.

Sun Patch Manager Enhancements

This feature was introduced in the Solaris Express 6/04 release.

The Sun Patch Manager tool (Patch Manager) is the standard tool for managing patches on Solaris systems. Use this tool to apply patches to Solaris systems.

You can access Patch Manager by using the `smpatch` command-line interface.

Patch Manager has been enhanced with these features:

- **PatchPro analysis engine** – Patch Manager now incorporates PatchPro functionality to automate the patch management process. This process includes performing patch analysis on systems, then downloading and applying the resulting patches. This automation functionality was previously available for the Solaris 9 release as a separate PatchPro product and is now part of the standard Solaris release. Solaris 9 users should refer to "Solaris Patch Update Feature" as described in *What's New in the Solaris 9 4/03 Operating Environment*.
- **Local-mode command-line interface** – The command-line interface, `smpatch`, can be used even when the Solaris WBEM services are not running on your system. This capability enables you to use `smpatch` to apply patches while your system is in single-user mode.

- **Patch list operations** – Patch Manager enables you to generate, save, edit, order, and resolve patch lists. These lists can be used to perform patch operations, such as downloading and applying patches.

You must install at least the Developer Software Support Group of the Solaris 10 release to use Sun Patch Manager.

For more information about how to use Patch Manager, see Chapter 19, "Managing Solaris Patches by Using Sun Patch Manager (Tasks)," in the *System Administration Guide: Basic Administration*. See also the smpatch(1M) man page.

New `psrinfo` Option to Identify Chip Multithreading Features

This feature was introduced in the Solaris Express 6/04 release and in the Solaris 9 12/03 release.

In the Solaris 10 OS, the psrinfo command has been modified to provide information about physical processors, in addition to information about virtual processors. This enhanced functionality has been added to identify chip multithreading (CMT) features. The new -p option reports the total number of physical processors that are in a system. Using the psrinfo -pv command lists all the physical processors that are in the system, as well as the virtual processors that are associated with each physical processor.

The default output of the psrinfo command continues to display the virtual processor information for a system.

For more information, see the psrinfo(1M) man page. For information about the procedures that are associated with this feature, see the *System Administration Guide: Advanced Administration*.

Enhancements to `pfiles` Tool

This feature was introduced in the Solaris Express 6/04 release.

The /proc file system has been enhanced to include file name information in the /proc/<pid>/path directory. This information is used by pfiles(1) to display file names for each file in the process. This enhancement creates an entirely new form of observability that provides new insight into process behavior. This new observability benefits both administrators and developers.

Solaris IP Filter

This feature was introduced in the Solaris Express 2/04 release.

Solaris IP Filter is a host-based firewall that is based on the open source IP Filter. Solaris IP Filter provides packet filtering and network address translation (NAT), based on a user-configurable policy. Packet filtering rules are configurable in either a stateful or stateless manner. All configuration and management of Solaris IP Filter is performed through a command-line interface.

For further information, see the ipfilter(5) man page. Also, refer to the ipf(1M), ipfs(1M), and ipfstat(1M) man pages. In addition, see the *System Administration Guide: IP Services*.

Core File Content Enhancements

The Solaris Express 12/03 release introduced new enhancements to the coreadm, gcore, and mdb utilities. These changes improve management of core files as described in the following three sections. Further enhancements to the coreadm command were made in the Solaris Express 1/04 release.

Specify Core File Content With coreadm

This feature was introduced in the Solaris Express 12/03 release. The -i and -I options have been further enhanced in the Solaris Express 1/04 release.

In the Solaris Express 12/03 release, the coreadm command lets you specify which parts of a process are present in the core file during a crash. You can see the system's configuration by running coreadm with no arguments.

You can specify the global core file content and the default per-process core file content by using the -G and -I options respectively. Each option requires a set of content specifier tokens. You can also set the core file content for individual processes by using the -P option. Core dumps that correspond to the global settings no longer honor the per-process, core file-size resource control.

In the Solaris Express 1/04 release, the -i and -I options to the coreadm command now apply to all processes whose core file settings are using the system-wide default. Use the -p and -P options to override the default.

For further information, see the coreadm(1M) man page.

gcore Core File Content

This feature was introduced in the Solaris Express 12/03 release.

The gcore utility creates core files from a running process without damaging that process. The gcore utility now supports variable core file content. Use the -c option to specify the content or the -p or -g options to force gcore to use the coreadm settings.

See the gcore(1) man page for further information.

mdb Supports Text and Symbol Tables in Core Files

This feature was introduced in the Solaris Express 12/03 release.

Text is now in core files by default. Also, symbol tables can now be in core files by default. The mdb utility has been updated to support this new core file data. This support means you can now debug your old core file without needing the original binary or the libraries that are linked to that file.

See the mdb(1) man page for further information.

System Management Agent

This feature was introduced in the Solaris Express 12/03 release.

The System Management Agent is a Simple Network Management Protocol (SNMP) agent that provides SNMPv1, SNMPv2c, and SNMPv3 functionality to the Solaris 10 environment. The agent is based on the Net-SNMP open source implementation, with some customizations for the Solaris environment. The agent has all the necessary base functionality that is required by an SNMP agent. The agent includes support for standard SNMP operations and numerous standard Management Information Bases (MIBs), including MIB-II, Host Resources MIB, and Notification MIB. Additionally, the agent supports the User-based Security Model (USM) and View-based Access Control Model (VACM), as well as AgentX.

The System Management Agent is configured to be the default SNMP agent, but coexists with the Solstice Enterprise Agents™ software in this release.

For further information, see the netsnmp(5) man page.

For further information, see also the *Solaris System Management Agent Administration Guide* and the *Solaris System Management Agent Developer's Guide*.

SPARC: 64-bit Package Changes for System Administrators

This feature was introduced in the Software Express pilot program.

In Solaris software, packaging has been simplified with most 32-bit components and 64-bit components being delivered in a single package. For further information, see "SPARC: 64-bit Package Changes" on page 45.

NIS-to-LDAP Transition Service

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/03 release.

The NIS-to-LDAP transition service can be used to enable a network transition from using NIS to using LDAP as the primary naming service. By using this transition service, administrators can utilize the Sun Java System Directory Server, which works with LDAP naming service clients.

For more information, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

Signed Packages and Patches

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/03 release.

The Solaris software enables you to securely download Solaris packages and patches that include a digital signature by using the updated `pkgadd` and `patchadd` commands. A package or a patch with a valid digital signature ensures that the package or patch has not been modified after the signature was applied to the package or patch.

In previous Solaris releases, you could only add signed patches to your system if you used the Solaris patch management tools with PatchPro 2.1.

Additional software management features in this Solaris release include the following:

- You can add a digital signature to a package with the updated `pkgtrans` command. For information about creating a signed package, see the *Application Packaging Developer's Guide*.
- You can download a package or patch from an HTTP or HTTPS server.

A signed package is identical to an unsigned package except for the signature. The package can be installed, queried, or removed with existing Solaris packaging tools. A signed package is also binary-compatible with an unsigned package.

Before you can add a package or patch with digital signatures to your system, you must set up a keystore with trusted certificates that are used to identify that the digital signature on the package or patch is valid.

For information about setting up the package keystore and adding signed packages or patches to your system, see the *System Administration Guide: Basic Administration*.

For information about booting and retrieving Solaris installation images from an HTTP or HTTPS server, see .

System V IPC Configuration

This feature was introduced in the Software Express pilot program.

In the Solaris software, all System V IPC facilities are either automatically configured, or the facilities can be controlled by resource controls. These facilities might be shared memory, message queues, or semaphores. Resource controls allow IPC settings to be made on a per-project or per-user basis on the local system or in a name service environment. In previous Solaris releases, IPC facilities were controlled by kernel tunables. You had to modify the `/etc/system` file and reboot the system to change the default values for these facilities. Because the IPC facilities are now controlled by resource controls, their configuration can be modified while the system is running. Many applications that previously required system tuning to function might now run without tuning because of increased defaults and automatic allocation of resources.

The following table identifies the now obsolete IPC tunables that are described in the *Solaris Tunable Parameters Reference Manual* and their replacement resource controls.

Resource Control	Obsolete Tunable	Old Default Value	Maximum Value	New Default Value
`process.max-msg-qbytes`	`msginfo_msgmnb`	4096	`ULONG_MAX`	65536
`process.max-msg-messages`	`msginfo_msgtql`	40	`UINT_MAX`	8192
`process.max-sem-ops`	`seminfo_semopm`	10	`INT_MAX`	512
`process.max-sem-nsems`	`seminfo_semmsl`	25	`SHRT_MAX`	512
`project.max-shm-memory`	`shminfo_shmmax`	0x800000	`UINT64_MAX`	1/4 of physical memory
`project.max-shm-ids`	`shminfo_shmmni`	100	2^{24}	128
`project.max-msg-ids`	`msginfo_msgmni`	50	2^{24}	128
`project.max-sem-ids`	`seminfo_semmni`	10	2^{24}	128

Obsolete parameters might be included in the /etc/system file on this Solaris system. If so, the parameters are used to initialize the default resource control values as in previous Solaris releases. However, using the obsolete parameters is not recommended.

The following related parameters have been removed. If these parameters are included in the /etc/system file on this Solaris system, the parameters are commented out.

- `semsys:seminfo_semmns`
- `semsys:seminfo_semvmx`
- `semsys:seminfo_semmnu`
- `semsys:seminfo_semaem`
- `semsys:seminfo_semume`
- `semsys:seminfo_semusz`
- `semsys:seminfo_semmap`
- `shmsys:shminfo_shmseg`
- `shmsys:shminfo_shmmin`
- `msgsys:msginfo_msgmap`
- `msgsys:msginfo_msgseg`
- `msgsys:msginfo_msgssz`
- `msgsys:msginfo_msgmax`

For more information about using the new resource controls, see "System V IPC and Other Resource Controls" on page 36.

Optional Interval for `netstat` Command

This feature was introduced in the Software Express pilot program.

The netstat command displays the contents of certain network-related data structures in various formats, depending on the options that you select. The -s, -i and -m options now support an "interval." With the specification of an optional interval, the output is continuously displayed in interval seconds until interrupted by the user. If a "count" is specified after "interval," the output iterates for "count" times only. The "interval" and "count" must both be positive integers.

For further information, see the netstat(1M) man page.

GMT-offset Quoted Time Zones

This feature was introduced in the Software Express pilot program.

The Solaris Install program installs a quoted time zone as the system default time zone, if a GMT-offset style time zone is selected. Previously, an unquoted time zone was installed.

Note that quoted time zones behave differently than unquoted time zones. When the time zone name is displayed by the date command, for example, quoted time zones display the offset from GMT. Unquoted time zones display only the string "GMT," without the offset from GMT.

Combine Data Address and Test Address for Single IP Network Multipathing Group

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

The use of a dedicated test IP address for failure detection on single-adapter IP network multipathing groups is no longer required. Sometimes, failover is not possible because only one network interface card (NIC) is in an IP network multipathing group. In this situation, you can now combine the test address and the data address. The in.mpathd daemon uses a data address to detect failure when a test address is not specified.

For further information on Solaris IP multipathing, see Part VI, "IPMP," in the *System Administration Guide: IP Services*. See also the in.mpathd(1M) man page.

LDAP Command Changes

These command changes were made in the Solaris Express 12/03 release.

Several LDAP commands are updated to include full SSL support and extended support for SASL. The modifications also provide support for managing smart referrals, using virtual list views (VLVs), and establishing stronger authentication when binding to the LDAP server.

This upgrade aligns the Solaris LDAP command functionality with Sun's LDAP directory server commands. All Solaris functionality is preserved for backward compatibility. The updated commands are ldapdelete, ldapmodify, ldapadd, ldapsearch, and ldapmodrdn.

Several changes have been made to the LDAP commands. Those changes include the following:

- The -M authentication option is obsolete. This option has been superseded by the stronger -o option. The -M option is now used for managing smart referrals.
- Search results are now displayed in LDAP Data Interchange Format (LDIF) by default. Use the -r option to display results in the old format, for backward compatibility.

See the ldapdelete(1), ldapmodify(1), ldapadd(1), ldapsearch(1), and ldapmodrdn(1) man pages for details.

For further information, see the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

Desktop Features

This section describes all desktop features in the Solaris 10 Operating System (OS) that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002. The Java Desktop System, Release 3 is an important addition to the Solaris 10 system.

Java Desktop System, Release 3

This feature was introduced in the Solaris Express 10/04 release.

The Sun Java Desktop System, Release 3 for the Solaris 10 OS combines open source software with Sun innovation to offer a comprehensive, fully integrated, and interoperable desktop environment with an accessible, modern user interface.

Comprehensive Environment

The Java Desktop System provides a comprehensive desktop environment that enables users to navigate and work effectively with the applications and documents on their desktop. The system includes the following:

- GNOME desktop environment
- StarOffice™ productivity suite
- Evolution mail, with calendar and tasks
- Mozilla™ browser and email client
- Complete set of software utilities and accessories

Intuitive User Interface

The Java Desktop System provides an intuitive desktop interface with familiar desktop themes and file manager views to enable users to work effectively and efficiently. Other features of the user interface include the following:

- Easy to learn and use, especially for users with Windows experience, featuring:
  - Familiar desktop themes for file management and desktop configuration

- Standard keyboard shortcut conventions
- Direct access to files and servers using the "Documents" and "Network Places" directories
- Familiar access to networked computers from a single "Network Places" directory
- Comprehensive user documentation and Help

Integrated Desktop

Desktop applications are well integrated in the Java Desktop System, providing interoperability between applications. Interoperability features include the following:

- Drag-and-drop and copy or paste text, graphics, and other elements between applications
- Universal printer selection of available printers on the network, to include printers on Windows networks
- Single point of entry for directory and file management and networked computers
- Direct access to content in data and files by single-click application launch for files contained in e-mail, Web pages, and file manager

Enterprise-level Interoperability

In addition to providing interoperability between applications in the desktop, the Java Desktop System provides interoperability at the enterprise level. Features include the following:

- Users can access data, files, and printers with both Windows and UNIX environments.
- The system can connect to existing backend systems, including Sun Java System Calendaring and Messaging servers and Active Directory, POP3, LDAP, and SMTP mail servers.

For further information about the desktop, see the *Java Desktop System Release 3 User Guide*. For further information about the applications included in the Java Desktop System, open Help in the desktop and select the specific application from the navigation pane.

Accessibility Features

The Java Desktop System incorporates many new accessibility features to support people with disabilities. Features in the Solaris 10 OS include enhanced keyboard and desktop navigation, which make the desktop accessible and customizable. The new assistive technologies included in the release are gnopernicus, a screen reader and screen magnifier, and a GNOME onscreen keyboard (GOK).

For further information, see the *Java Desktop System Release 3 Accessibility Guide*.

Mozilla 1.7

Mozilla 1.4 was introduced in the Solaris Express 4/04 release. Mozilla 1.7 was introduced in the Solaris Express 10/04 release.

Mozilla is Sun's new standard browser for the Solaris Operating System, delivering functionality that enables industry-standard web interaction.

Mozilla includes tools for the following:

- Web browsing
- Managing email
- Communicating with colleagues
- Participating in discussion groups
- Creating dynamic web pages

Mozilla includes the following new features:

- Non-Standard web page support
- Profile roaming support
- Junk-mail filtering
- Enhanced pop-up window blocking
- Easier bookmark management

For more information about Mozilla, go to `http://www.sun.com/software/solaris/browser/`.

System Resources Enhancements

This section describes all system resources enhancements in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002. The Solaris 10 OS includes a feature of key importance to Solaris users, the Solaris Zones partitioning technology.

Solaris Zones Software Partitioning Technology

This feature was introduced in the Solaris Express 2/04 release. In the Solaris Express 7/04 release, new functionality for Zones has been added.

The Solaris Zones software partitioning technology, a component of the Solaris Containers environment, is a software partitioning technology used to virtualize operating system services and provide an isolated and secure environment for running applications. A zone is a virtualized operating system environment created within a single instance of the Solaris Operating System. Zones basically provide the standard Solaris interfaces and application environment, and do not include a new ABI or API that would require applications to be ported.

Each zone can provide a customized set of services. Zones are ideal for environments that consolidate multiple applications on a single server. Resource management features can be used within zones to further control how applications use available system resources.

A zone can be thought of as a box. One or more applications can run in this box without affecting the rest of the system. This isolation prevents processes that are running in one zone from monitoring or interfering with processes that are running in other zones. Even a process with superuser credentials that is running inside a zone cannot view or affect activity in other zones.

The single instance of the Solaris Operating System is the global zone. The global zone is both the default zone for the system and the zone used for system-wide administrative control. One or more non-global zones can be created by an administrator working in the global zone. Once created, these non-global zones can be administered by individual zone administrators. The privileges of a zone administrator are confined to a non-global zone.

Non-global zones provide isolation at almost any level of granularity you require. A zone does not need a dedicated CPU, a physical device, or a portion of physical memory. These resources can either be multiplexed across several zones running within a single domain or system, or allocated on a per-zone basis using the resource management features available in the operating system. Even a small uniprocessor system can support multiple zones running simultaneously.

To achieve process isolation, a process can see or signal only those processes that exist in the same zone.

Basic communication between zones is provided by giving each zone at least one logical network interface. Applications running in different zones on the same system can bind to the same network port by using the distinct IP addresses associated with each zone or by using the wildcard address. An application running in one zone cannot observe the network traffic of another zone. This isolation is maintained even though the respective streams of packets travel through the same physical interface.

Each zone is given a portion of the file system hierarchy. Because each zone is confined to its subtree of the file system hierarchy, a workload running in a particular zone cannot access the on-disk data of another workload running in a different zone.

Files used by naming services reside within a zone's own root file system view. Thus, naming services in different zones are isolated from one other and can be configured differently.

For information about how to configure and use zones on your system, see the *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*.

System V IPC and Other Resource Controls

The resource controls (rctls) feature was introduced in the Solaris 9 software. Additional resource controls were introduced in the Software Express pilot program, and in the Solaris Express 3/04, 8/04, and 10/04 releases.

New project-based and new process-based resource controls have been added. Resource controls provide a mechanism for controlling how applications use system resources.

In the Software Express pilot program, the following resource controls were added. These resource controls affect the System V IPC (interprocess communication) facilities, which include shared memory, message queues, and semaphores.

- `project.max-shm-ids`
- `project.max-sem-ids`
- `project.max-msg-ids`
- `project.max-shm-memory`
- `process.max-sem-nsems`
- `process.max-sem-ops`
- `process.max-msg-qbytes`

See "System V IPC Configuration" on page 30 for more information.

In the Solaris Express 3/04 release, the following event port resource controls were added:

- `project.max-device-locked-memory`
- `project.max-port-ids`
- `process.max-port-events`

In the Solaris Express 8/04 release, the following new resource controls have been added:

- `project.max-lwps`
- `project.max-tasks`

In the Solaris Express 10/04 release, the `project.max-contracts` resource control was added.

In the Solaris Express 11/04 release, the `process.max-crypto-memory` cryptographic resource control was added.

Information about resource controls is provided in Chapter 6, "Resource Controls (Overview)," in the *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*. This chapter includes the following:

- A current list of available resource controls
- A general description of resource controls
- Resource control configuration information

Information about configuring resource controls is also available in the `project`(4) man page.

New Solaris Project and Resource Management Command Functionality

This feature was introduced in the Solaris Express 7/04 release.

Enhancements to project database and resource control commands that were introduced in the Solaris Express 7/04 release include the following:

- Scaled value and unit modifier support for resource control values and commands
- Improved validation and easier manipulation of the project attributes field

  See the project(4) man page.

- Revised output format and new options for the prctl and projects commands

  See the prctl(1) and projects(1) man pages.

For information about the changes introduced, see the following chapters in *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*:

- Chapter 2, "Projects and Tasks (Overview)"
- Chapter 3, "Administering Projects and Tasks"
- Chapter 6, "Resource Controls (Overview)"
- Chapter 7, "Administering Resource Controls (Tasks)"

Note that information about setting the user's default project through the useradd, usermod, and passmgmt commands has been updated in the guide. These commands are documented in the useradd(1M), usermod(1M), and passmgmt(1M) man pages.

See also the following man pages:

- prctl(1)
- projects(1)
- projadd(1M)
- projdel(1M)
- projmod(1M)
- rctladm(1M)
- setrctl(2)
- rctlblk_set_value(3C)
- setproject(3PROJECT)
- project(4)

Dynamic Resource Pools

This feature was introduced in the Solaris Express 1/04 release.

Dynamic resource pools (DRPs) provide a mechanism for adjusting each pool's resource allocation in response to system events and application load changes on systems that have resource pools enabled. Adjustments are automatically made to maintain the system performance goals specified by an administrator. Changes that are made to the configuration are logged.

These features are primarily enacted through the resource controller poold, a system daemon that is active when dynamic resource allocation is required. Periodically, poold examines the load on the system and determines whether intervention is required to enable the system to maintain the specified resource utilization goals. The daemon takes corrective action if possible, or the condition is logged.

For further information, see the following chapters in the *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*:

- Chapter 12, "Dynamic Resource Pools (Overview)"
- Chapter 13, "Administering Dynamic Resource Pools (Tasks)"
- Chapter 14, "Resource Management Configuration Example"

See also the following man pages:

- pooladm(1M)
- poolbind(1M)
- poolcfg(1M)
- poold(1M)
- poolstat(1M)
- libpool(3LIB)

Extended Accounting Subsystem Enhancements

Extended accounting was introduced when the Solaris 9 Operating System was originally distributed in May 2002. The Software Express pilot program introduced flow accounting enhancements and a Perl interface.

Note – For Solaris 9 users, flow accounting enhancements were introduced in the Solaris 9 9/02 release. The Perl interface was introduced in the Solaris 9 4/03 release.

You can use extended accounting processes in conjunction with the flow accounting module for IPQoS. For information about IPQoS, see Part VII, "IP Quality of Service (IPQoS)," in the *System Administration Guide: IP Services*.

For information about the extended accounting feature, see Chapter 5, "Administering Extended Accounting (Tasks)," in the *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*.

A Perl interface to libexacct is available in the Software Express releases. This interface enables you to create Perl scripts that can read the accounting files which are produced by the exacct framework. You can also create Perl scripts that write exacct files. The new interface is functionally equivalent to the underlying C API.

You can use the Perl interface to record system resource consumption on a task or process basis. Or, you can record consumption on the basis of selectors provided by the IPQoS flowacct module.

For further information, see the following man pages:

- Exacct(3PERL)
- Exacct::Catalog(3PERL)
- Exacct::File(3PERL)
- Exacct::Object(3PERL)
- Exacct::Object::Group(3PERL)

- `Exacct::Object::Item`(3PERL)
- `Kstat`(3PERL)
- `Project`(3PERL)
- `Task`(3PERL)

For information on how to configure and use extended accounting with enhancements described in this section, see Chapter 4, "Extended Accounting (Overview)," in the *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*.

Physical Memory Control Using the Resource Capping Daemon

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/03 release.

Physical memory control that uses the resource capping daemon is an optional feature. The resource capping daemon `rcapd` regulates the consumption of physical memory by processes that run in projects that have defined resource caps. Associated utilities provide mechanisms for administering the daemon and reporting related statistics.

For additional information, see Chapter 10, "Physical Memory Control Using the Resource Capping Daemon (Overview)," in the *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*.

See also the man pages `rcapstat`(1), `rcapadm`(1M), `project`(4), and `rcapd`(1M).

Java 2 Platform, Standard Edition 5

This section describes Java platform features that were introduced in the Solaris Express 8/04 release.

Java 2 Platform, Standard Edition (J2SE™) 5 is a significant release including many new features and updates while preserving compatibility and stability. For more details about the features described in this section, see `http://java.sun.com/j2se/5.0`.

New Additions to the Java Programming Language

In the Solaris Express 8/04 release, the following features are now available to developers who use the Java language:

- The ability to declare type-safe generic types.
- Automatic conversion between primitive types and their object counterparts, referred to as "autoboxing" and "autounboxing."
- The ability to create type-safe enumerated types, more specifically, an enhanced "for loop."
- The ability to import a static namespace, an annotation API for use with packages, fields, methods, and classes.
- Permission to use subclassed return types, covariant returns, and variable argument lists. The javac compiler enables these features by default.

The Java programming language changes reduce the complexity and the amount of additional code that is required for common Java programming tasks. These changes also reduce the risk of runtime errors that a developer might introduce.

Monitoring and Manageability Enhancements for Java Platform

In the Solaris Express 8/04 release, the Java Virtual Machine (JVM) internals can now be monitored by industry-standard consoles that use JMX or SNMP protocols. Both mechanisms are disabled by default and are preconfigured to provide low memory notifications. The JMX mechanism also allows properties such as the logging level to be dynamically configured at runtime.

For further information, see
`http://java.sun.com/j2se/1.5.0/docs/guide/management/index.html`.

Performance and Scalability Improvements for Java Platform

In the Solaris Express 8/04 release, J2SE 5 offers enhanced performance and scalability for developers who use the Java platform.

Automatic Tuning for Large Server Machines

Machines with more than 2 Gbytes of main memory and more than 2 processors automatically select an enhanced server performance mode. This mode delivers significant performance, especially for J2EE™ based application servers. This enhanced performance mode uses the "hotspot" server compiler with a parallel "garbage collector," and with appropriately tuned initial and maximum memory sizes.

For further information, see
`http://java.sun.com/j2se/1.5.0/docs/guide/vm/gc-ergonomics.html`.

Reduced Startup Time

Desktop applications benefit from the addition of class data sharing. During installation, this feature creates a special preprocessed archive of commonly used Java classes. This same archive is loaded when the JVM starts, thus reducing the processing time. This archive is also available to subsequent Java applications on that same machine.

OpenGL Hardware Acceleration

The JVM can accelerate JFC/Swing and Java 2D applications if an OpenGL device is available. This acceleration is automatic but disabled by default. This acceleration improves the runtime performance of many graphic-intensive programs.

XML 1.1 and Namespace, XSLTC, SAX 2.0.2, and DOM Level 3, Java based APIs

This feature was introduced in the Solaris Express 8/04 release.

Developers can now use the later XML 1.1 syntax. Developers can process XML by using the updates to the existing event-based XML processor, SAX 2.0.2. Developers can also use the newer document-based processor, DOM Level 3. For handling XML transforms, the fast XSLT compiler (XSLTC) is available.

New Default Swing Look and Feel

This feature was introduced in the Solaris Express 8/04 release.

The default cross-platform "look and feel" has been updated. Titled Ocean, this new "look and feel" is enabled by default. In addition, a GNOME-based GTK "look and feel" is now fully available for developers and users.

Diagnostic Tools

This feature was introduced in the Solaris Express 8/04 release.

In addition to using the monitoring protocols, administrators and developers can use new command-based diagnostic tools. Tools are available for the following tasks:

- Performance analysis – jstat
- Fatal error handling – XX:OnError
- Core file analysis – jdi connectors
- JVM internals tools – jstack, jmap, jinfo

In addition, Java extensions are available for the DTrace tracing facility.

Unicode Support

In the Solaris Express 8/04 release, Unicode 4.0-based supplementary characters are supported in J2SE 5.

Developers can now use Unicode-based supplementary characters, represented as pairs of Java characters, with the Java language. When using the string-based and text-based APIs, no additional effort is required.

For further information, see `http://java.sun.com/j2se/1.5.0/docs/`.

Note – Previously, Unicode 3.2 support was available in the Software Express pilot program. For Solaris 9 users, Unicode 3.2 support was introduced in the Solaris 9 8/03 release.

Installation Enhancements

This section describes all installation enhancements in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002. The installation unification enhancements are of particular importance.

Solaris Installation Changes Including Installation Unification

Introduced in the Solaris Express 6/04 release, several changes in the installation of the Solaris OS provide a more simplified and unified installation experience.

The changes include the following:

- This release has one installation DVD and several installation CDs. The Solaris 10 Operating System DVD includes the content of all the installation CDs.

  - **Solaris Software 1** – This CD is the only bootable CD. From this CD, you can access both the Solaris installation graphical user interface (GUI) and the console-based installation. This CD also enables you to install selected software products from both the GUI and the console-based installation.

  - **Other Solaris Operating System CDs** – These CDs contain the following:

    - Solaris packages that the software prompts you to install if necessary
    - ExtraValue software that includes supported and unsupported software
    - Installers
    - Localized interface software and documentation

- The Solaris Installation CD no longer exists.

- For both CD and DVD media, the GUI installation is the default (if your system has enough memory). However, you can specify a console-based installation with the text boot option.

- The installation process has been simplified, enabling you to select the language support at boot time, but select locales later.

Note – The (noninteractive) Solaris custom JumpStart™ installation method has not changed.

To install the OS, simply insert the Solaris Software - 1 CD or the Solaris Operating System DVD and type one of the following commands.

- For the default GUI installation (if system memory permits), type **`boot cdrom`**.
- For the console-based installation, type **`boot cdrom - text`**.

For instructions about how to install the Solaris OS by using CD or DVD media with the new text boot option, see *Solaris 10 Installation Guide: Basic Installations*. For changes to setting up an installation server with CD media, see *Solaris 10 Installation Guide: Network-Based Installations*.

Accessing the GUI or Console-based Installations

You can choose to install the software with a GUI or with or without a windowing environment. Given sufficient memory, the GUI is displayed by default. If the memory is insufficient for the GUI, other environments are displayed by default. You can override defaults with the nowin or text boot options. But, you are limited by the amount of memory in your system or by installing remotely. Also, if the Solaris

installation program does not detect a video adapter, the program is automatically displayed in a console-based environment. The following table describes these environments and lists minimal memory requirements for displaying them.

Minimal Memory	Type of Installation
128–383 MB	Console-based
384 MB or greater	GUI-based

Detailed descriptions for each installation option are as follows:

Installation with 128–383 MB minimal memory
This option contains no graphics, but provides a window and the ability to open other windows. This option requires a local or remote DVD-ROM or CD-ROM drive or network connection, video adapter, keyboard, and monitor. If you install by using the text boot option and have enough memory, you are installing in a windowing environment. If you are installing remotely through a tip line or by using the nowin boot option, you are limited to the console-based installation.

Installation with 384 MB memory or greater
This option provides windows, pull-down menus, buttons, scrollbars, and iconic images. A GUI requires a local or remote DVD-ROM or CD-ROM drive or network connection, video adapter, keyboard, and monitor.

Custom JumpStart Installation Package and Patch Enhancements

This feature was introduced in the Solaris Express 3/04 release.

When installing and upgrading the Solaris OS by using the custom JumpStart installation method, new customizations enable the following:

- A Solaris Flash installation with additional packages

  The custom JumpStart profile package keyword has been enhanced to enable installing a Solaris Flash archive with additional packages. For example, you can install the same base archive on two machines, but add a different set of packages to each machine. These packages do not have to be a part of the Solaris OS distribution.

- An installation with additional packages that might not be part of the Solaris distribution

  The package keyword has also been enhanced to enable an installation with a package that is not part of the Solaris distribution. You no longer need to write a postinstallation script to add extra packages.

- An installation with the ability to install Solaris OS patches

  The new custom JumpStart profile patch keyword enables the installation of Solaris OS patches. This feature allows the installation of a list of patches that are specified in a patch file.

For further information, see *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

x86: Specifying Boot Properties With `add_install_client` Command

This feature was introduced in the Solaris Express 12/03 release and in the Solaris 9 4/04 release.

The `add_install_client` command now enables you to set boot properties for Solaris x86 clients during a PXE network boot. The `-b` option enables you to perform the following tasks with the `add_install_client` command.

- You can specify an alternate console to use during your network installation.
- You can specify the device to use as the network boot device during the installation.
- You can instruct the client to perform a fully automated custom JumpStart installation.

For more information, see the `install_scripts`(1M) man page, the `eeprom`(1M) man page, and *Solaris 10 Installation Guide: Network-Based Installations*.

Configuring Multiple Network Interfaces During Installation

This feature was introduced in the Solaris Express 12/03 release.

The Solaris installation programs enable you to configure multiple interfaces during your installation. You can preconfigure these interfaces in the `sysidcfg` file for your system. Or, you can configure multiple interfaces during the installation.

For more information, see the `sysidtool`(1M) man page and the `sysidcfg`(4) man page. See also the *Solaris 10 Installation Guide: Network-Based Installations*.

Replacement of Solaris Installation Command-Line Interface

This feature was introduced in the Solaris Express 9/03 release.

The command-line interface of the Solaris installation program has been replaced to improve the usability of console-based installations. All console-based installations use the forms-based program to install the Solaris Operating System.

See the `install-solaris`(1M) man page for further information.

SPARC: 64-bit Package Changes

This feature was introduced in the Software Express pilot program.

In previous Solaris releases, the Solaris software was delivered in separate packages for 32-bit components and 64-bit components. Now, packaging has been simplified with most 32-bit components and 64-bit components being delivered in a single package. The combined packages retain the names of the original 32-bit packages, and the 64-bit packages are no longer delivered.

The removal of the 64-bit packages simplifies installation and increases performance:

- Reduces the number of packages, which simplifies Custom JumpStart scripts that contain lists of packages
- Simplifies the packaging system with only one package that groups software functions
- Reduces installation time because fewer packages are installed

The 64-bit packages are renamed with the following conventions:

- If a 64-bit package has a 32-bit counterpart, the 64-bit package is named with the 32-bit package name. For example, a 64-bit library such as `/usr/lib/sparcv9/libc.so.1` previously would have been delivered in `SUNWcslx`, but now is delivered in `SUNWcsl`. The 64-bit `SUNWcslx` package is no longer delivered.
- If a package does not have a 32-bit counterpart, the "x" suffix is removed from the name. For example, `SUNW1394x` becomes `SUNW1394`.

This change means that you might need to modify your custom JumpStart script or other package installation scripts to remove references to the 64-bit packages.

WAN Boot Installation Method

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/03 release.

The Solaris software now enables you to boot and install software over a wide area network (WAN) by using HTTP. The WAN boot installation method enables you to install the Solaris release on systems over a large public network where the network infrastructure might be untrustworthy. You can use WAN boot with new security features to protect data confidentiality and installation image integrity.

The WAN boot installation method enables you to transmit an encrypted Solaris Flash archive over a public network to a remote client. The WAN boot programs then install the client system by performing a custom JumpStart installation. To protect the integrity of the installation, you can use private keys to authenticate and encrypt data. You can also transmit your installation data and files over a secure HTTP connection by configuring your systems to use digital certificates.

For more information about this feature, see the *Solaris 10 Installation Guide: Network-Based Installations*.

Creating RAID-1 Volumes (Mirrors) With Custom JumpStart

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/04 release.

The custom JumpStart installation method now enables you to create RAID-1 volumes (mirrors) during the installation of the Solaris Operating System. By mirroring file systems, you can protect your system by duplicating data over two physical disks. If one of the mirrored disks fails, the system data is still accessible from the second mirrored disk.

In JumpStart, the following new custom profile keywords and values enable you to create mirrored file systems.

- The new `filesys` keyword value mirror creates a mirror. Then you can designate specific slices as single-slice concatenations to attach to the mirror.
- The new `metadb` profile keyword enables you to create the required state database replicas.

For more information, see *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Solaris Live Upgrade 2.1

This feature was introduced in the Software Express pilot program and in the Solaris 9 8/03 release.

Solaris Live Upgrade provides a method of upgrading a system while the system continues to operate. While your current boot environment is running, you can duplicate the boot environment, then upgrade the duplicate. Or, rather than upgrading, you can install a Solaris Flash archive on a boot environment. The original system configuration remains fully functional and unaffected by the upgrade or installation of an archive. When you are ready, you can activate the new boot environment by rebooting the system. If a failure occurs, you have a safety net. You can quickly revert to the original boot environment with a simple reboot. Thus, you eliminate the normal downtime of the test and evaluation process.

New features in Solaris Live Upgrade 2.1 provide the following new functionality.

- Solaris Live Upgrade uses Solaris Volume Manager technology to create a duplicate boot environment that contains file systems with RAID-1 volumes (mirrors). The mirror provides data redundancy for any file systems, including the root (/) file system. With the `lucreate` command, you can create mirrored file systems that contain up to three submirrors.
- With the `lucreate` command, you can now exclude some files and directories that would normally be copied from the original boot environment. If you have excluded a directory, you can also include specified files and subdirectories under that directory.

For further information, see *Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning*.

Custom JumpStart Installation Method Creates New Boot Environment

This feature was introduced in the Software Express pilot program.

You can now use the JumpStart installation method to create an empty boot environment when you install the Solaris Operating System. The empty boot environment can then be populated with a Solaris Flash archive for later use.

For further information, see *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Solaris Flash Archives

The Solaris Flash features that are described in this section were introduced in the Software Express pilot program.

The Solaris Flash installation feature enables you to use a single reference installation of the Solaris software on a system. This system is called the master system. Then, you can replicate that installation on multiple systems, which are called clone systems. The installation is an initial installation that overwrites all files on the clone system.

Solaris Flash Differential Archives and Configuration Scripts

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

The Solaris Flash installation feature provides new enhancements for the Solaris software.

- A Solaris Flash installation can now update a clone system with minor changes. If you have a clone system and want to update it, you can create a differential archive that contains only the differences between two images, the original master image and an updated master image. When you update a clone system with a differential archive, only the files that are specified in the differential archive are changed. The installation is restricted to clone systems that contain software which is consistent with the original master image. You use the custom JumpStart installation method to install a differential archive on a clone system. Or, you can use Solaris Live Upgrade to install a differential archive on a duplicate boot environment.

- Special scripts can now be run for configuration of the master or clone or can be run to validate the archive. These scripts enable you to do the following tasks.

    - Configure applications on clone systems. You can use a custom JumpStart script for some uncomplicated configurations. For more complicated configurations, special configuration file processing might be necessary on the master system or before or after installation on the clone system. Also, local preinstallation and postinstallation scripts can reside on the clone. These scripts can protect local customizations from being overwritten by the Solaris Flash software.

    - Identify nonclonable, host-dependent data that enables you to make the flash archive host independent. Host independence is enabled by modifying such data or excluding the data from the archive. An example of host-dependent data is a log file.

    - Validate software integrity in the archive during creation.

    - Validate the installation on the clone system.

For further information, see *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*. This guide also includes information on how to use Solaris Live Upgrade to install a differential archive.

Customizing Content in a Solaris Flash Archive

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

The `flarcreate` command is used to create a Solaris Flash archive. This command has been updated with new options that increase your flexibility to define archive contents when creating an archive. You now can exclude more than one file or directory. From an excluded directory, you can add back a subdirectory or file. This feature is useful when you want to exclude large data files that you do not want cloned.

For information about how to use these options, see *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*.

Note – Solaris Flash was formerly Web Start Flash.

Command-Line Interface Enhancements to the Solaris Product Registry

These enhancements were introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

The `prodreg` command has been updated to include functionality that is similar to the Solaris Product Registry graphical user interface. You can now use the following `prodreg` subcommands on the command line or in administration scripts to perform a variety of tasks.

- `browse` – The `browse` subcommand enables you to view registered software in a terminal window. By repeating the `browse` subcommand, you can navigate through the directory hierarchy of registered software.
- `info` – The `info` subcommand enables you to view information about registered software.

  You can use the `info` subcommand to identify the following specifics.

  - Installation location of the software
  - Other software that the specified software requires
  - Other software that depends on the specified software
  - Software that has been damaged by the removal of packages that the software requires
- `unregister` – The `unregister` subcommand removes software installation information from the Solaris Product Registry. If you remove software from your system without properly uninstalling the software from the Registry, you can use

the `prodreg unregister` command to clean up the obsolete entries in the Solaris Product Registry.

- `uninstall` – The `uninstall` subcommand enables you to remove registered software from your system by launching that software's uninstallation program.

For more information, see the `prodreg`(1M) man page and the *System Administration Guide: Basic Administration*.

Reduced Networking Software Group

This feature was introduced in the Software Express pilot program.

You can now create a more secure system with fewer enabled network services by selecting or specifying the Reduced Networking software group (SUNWCrnet) during your installation. The Reduced Networking software group provides system administration utilities and a multiuser text-based console. SUNWCrnet enables the system to recognize network interfaces. During the installation, you can customize your system's configuration by adding software packages and by activating network services as needed.

For further information, see *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Modifying Disk Partition Tables by Using a Virtual Table of Contents (VTOC)

This feature was introduced in the Software Express pilot program.

The Solaris installation program now enables you to load existing slices from the virtual table of contents (VTOC.) You can now preserve and use the system's existing disk slice tables during your installation, rather than using the installer's default disk layout.

x86: Change in Default Boot-Disk Partition Layout

This feature was introduced in the Software Express pilot program.

A new feature in the Solaris installation program is a boot-disk partition layout. This layout, by default, accommodates the Service partition on Sun x86 based systems. This installation program enables you to preserve an existing Service partition.

The new default includes the following partitions.

- First partition – Service partition (existing size on system)
- Second partition – x86 boot partition (approximately 11 Mbytes)
- Third partition – Solaris Operating System partition (remaining space on the boot disk)

If you want to use this default layout, select Default when the Solaris installation program asks you to choose a boot-disk layout.

Note – If you install the Solaris OS for x86 based systems on a system that does not currently include a Service partition, the Solaris installation program does not create a new Service partition. If you want to create a Service partition on your system, first use your system's diagnostic CD to create the Service partition. After the Service partition is created, then install the Solaris Operating System.

For information about how to create the Service partition, see your hardware documentation.

For more information, see *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Support for Lightweight Directory Access Protocol (LDAP) Version 2 Profiles

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

The Solaris installation programs now support LDAP Version 2 profiles. These profiles enable you to configure your system to use a proxy credential level. During the Solaris installation programs, you can specify the LDAP proxy-bind distinguished name and proxy-bind password. With any installation method, you can preconfigure LDAP before installation by using the `proxy_dn` and `proxy_password` keywords in the `sysidcfg` file.

For information, see *Solaris 10 Installation Guide: Network-Based Installations*.

Security Enhancements

This section describes security enhancements in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002. "Process Rights Management" on page 52 and "Reduced Networking Software Group" on page 50 are of particular importance.

In addition to the security features described in this section, see also the following security-related feature descriptions in the Developer Tools section and the Installation section:

- "WAN Boot Installation Method" on page 46
- "Reduced Networking Software Group" on page 50
- "New Mechanisms for the Solaris Cryptographic Framework" on page 93
- "Retail and Nonretail Options for Providers in Solaris Cryptographic Framework" on page 93
- "Simple Authentication and Security Layer for Developers" on page 96
- "SPNEGO Pseudo-Mechanism for GSS-API Applications" on page 100
- "Enhanced `crypt()` Function for Software Developers" on page 101
- "Smart Card Terminal Interfaces" on page 102
- "Smartcard Middleware APIs" on page 102

Signing ELF Objects

This feature was introduced in the Solaris 10 release.

The libraries and executable files in the Solaris 10 release include digital signatures that can be used to verify the integrity of these files. The digital signature provides a way to detect any accidental change or deliberate tampering with the executable content of the file.

Plug-ins for the Solaris Cryptographic Framework are automatically verified when loaded by the system. The `elfsign` command can be used manually to verify any signed file. Developers and administrators can also use `elfsign` to sign their own code.

For further information, see the `elfsign`(1) man page.

Process Rights Management

This feature was introduced in the Software Express pilot program.

In the Solaris software, administrative tasks that previously required superuser rights are now protected by process rights management. Process rights management uses privileges to restrict processes at the command, user, role, or system level. A privilege is a discrete right that a process requires to perform an operation. The system restricts processes to only those privileges that are required to perform the current task. Therefore, fewer `root` processes are vulnerable to exploitation. The number of `setuid` programs has been greatly reduced.

As installed, the Software Express releases and the Solaris 10 release are completely compatible with previous releases of the Solaris Operating System in terms of the privileges enhancements. Unmodified programs that run as `root` run with all privileges.

Device Protection – Devices are protected with a security policy. The policy is enforced with privileges. Therefore, the permissions on a device file do not fully determine the device's availability. Privileges might also be required to operate the device.

System interfaces that were protected by UNIX permissions are now protected by privileges. For example, members of the group `sys` are no longer automatically allowed to open the `/dev/ip` device. Processes that are running with the `net_rawaccess` privilege can access the `/dev/ip` device. When the system boots, access to all devices is restricted until the `devfsadm` command runs during the boot sequence. The initial policy is as strict as possible. The policy prevents all users except the superuser from initiating connections.

See the following man pages for more information:

- `getdevpolicy`(1M)
- `ppriv`(1)
- `add_drv`(1M)

- `update_drv(1M)`
- `rem_drv(1M)`
- `devfsadm(1M)`

Processes that need to retrieve Solaris IP MIB information should open `/dev/arp` and push the "tcp" and "udp" modules. No privileges are required. This method is equivalent to opening `/dev/ip` and pushing the "arp", "tcp" and "udp" modules. Because opening `/dev/ip` now requires a privilege, the `/dev/arp` method is preferred.

For further information, see the following sections in the *System Administration Guide: Security Services*:

- "Using Roles and Privileges (Overview)"
- "Privileges (Overview)"
- "Privileges (Tasks)"

Changes to PAM for the Solaris 10 OS

A new `pam_deny` module was added in the Software Express pilot program and enhanced in the Solaris Express 6/04 release. The module can be used to deny access to named PAM services. By default, the `pam_deny` module is not used. For more information, see the `pam_deny(5)` man page.

The Solaris 10 software includes the following changes to the PAM framework.

- The `pam_authtok_check` module now allows for strict password checking that uses new tunables in the `/etc/default/passwd` file. The new tunables define the following:

  - A list of comma-separated dictionary files that are used for checking common dictionary words in a password

  - The minimum differences that are required between a new password and an old password

  - The minimum number of alphabetic and nonalphabetic characters that must be used in a new password

  - The minimum number of uppercase and lowercase letters that must be used in a new password

  - The number of allowable consecutive repeating characters

  - The number of digits that must be used in the new password

  - Whether whitespaces are allowed in the new password

- The `pam_unix_auth` module implements account locking for local users. Account locking is enabled by the `LOCK_AFTER_RETRIES` tunable in `/etc/security/policy.conf` and the `lock_after-retries` key in `/etc/user_attr`.

- A new `binding` control flag has been defined. If the PAM module is successful and no preceding modules that are flagged as `required` have failed, then PAM skips the remaining modules and the authentication request succeeds. However, if

a failure is returned, PAM records a required failure and then continues processing the stack. This control flag is documented in the `pam.conf`(4) man page.

- The `pam_unix` module has been removed and replaced by a set of service modules of equivalent or greater functionality. Many of these modules were introduced in the Solaris 9 system. Here is a list of the replacement modules:

  - `pam_authtok_check`
  - `pam_authtok_get`
  - `pam_authtok_store`
  - `pam_dhkeys`
  - `pam_passwd_auth`
  - `pam_unix_account`
  - `pam_unix_auth`
  - `pam_unix_cred`
  - `pam_unix_session`

- The functionality of the `pam_unix_auth` module has been split into two modules. The `pam_unix_auth` module now verifies that the password is correct for the user. The new `pam_unix_cred` module provides functions that establish user credential information.

- Additions to the `pam_krb5` module have been made to manage the Kerberos credentials cache by using the PAM framework. See "Kerberos Enhancements" on page 57.

`pam_ldap` Changes

The following `pam_ldap` changes were introduced in the Solaris Express 10/04 release, except for the account management feature. This management feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release. See the `pam_ldap`(5) man page for more information about these changes.

- The previously supported `use_first_pass` and `try_first_pass` options are obsolete as of this Solaris 10 software release. These options are no longer needed. The options can safely be removed from `pam.conf`, and are silently ignored.

- Password prompting must be provided for by stacking `pam_authtok_get` before `pam_ldap` in the authentication and password module stacks, and by including `pam_passwd_auth` in the `passwd_service_auth` stack.

- The previously supported password update function is replaced in this release by the use of `pam_authtok_store` with the `server_policy` option.

- The `pam_ldap` account management feature strengthens the overall security of the LDAP Naming Service. Specifically, the account management feature does the following:

  - Allows for tracking password aging and expiration

  - Prevents users from choosing trivial or previously used passwords

  - Warns users if their passwords are about to expire

  - Locks out users after repeated login failures

- Prevents users other than the authorized system administrator from deactivating initialized accounts

---

Note – A clean, automated update cannot be provided for the changes in the previous list. Therefore, an upgrade to a Solaris 10 or compatible release cannot automatically update the existing `pam.conf` file to reflect the `pam_ldap` changes. If the existing `pam.conf` file contains a `pam_ldap` configuration, the CLEANUP file notifies you after the upgrade. Examine the `pam.conf` file and modify it, as needed.

See the following man pages for further information:

- `pam_passwd_auth`(5)
- `pam_authtok_get`(5)
- `pam_authtok_store`(5)
- `pam.conf`(4)

For further information about Solaris naming and directory services, see the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*. For information about Solaris security features, see the *System Administration Guide: Security Services*.

Solaris Secure Shell Enhancements

This feature description was introduced in the Solaris Express 10/04 release.

The following enhancements to Solaris Secure Shell are available in the Solaris 10 OS:

- Solaris Secure Shell is based on OpenSSH 3.5p1. The Solaris implementation also includes features and bug fixes from versions prior to OpenSSH 3.8p1.
- Solaris Secure Shell now supports the use of the GSS-API for user and host authentication that uses Kerberos V.

  PAM support, including password-aging support, has been improved.
- The default value of X11Forwarding is yes in the `/etc/ssh/sshd` config file.
- The ARCFOUR and AES128-CTR ciphers are now available. ARCFOUR is also known as RC4. The AES cipher is AES in counter mode.
- For further enhancements, see the description for "sshd Daemon and /etc/default/login" on page 56.

For further information about security in the Solaris 10 OS, see the *System Administration Guide: Security Services*.

OpenSSL and OpenSSL PKCS#11 Engine

This feature was introduced in the Solaris Express 8/04 release.

This release of Solaris includes the OpenSSL libraries and commands in `/usr/sfw`.

This release also includes an OpenSSL Engine interface to PKCS#11 so that OpenSSL consumers can access hardware and software cryptographic providers from the Solaris Cryptographic Framework.

Note – Because of cryptographic import restrictions in some countries, symmetric key cryptographic algorithms are limited to 128–bit if the SUNWcry package is not installed. The SUNWcry package is not included with the Solaris software. This package is available instead as a separate controlled download.

`sshd` Daemon and `/etc/default/login`

This feature was introduced in the Solaris Express 10/04 release.

The `sshd` daemon uses the variables in `/etc/default/login` and the `login` command. The `etc/default/login` variables can be overridden by values in the `sshd_config` file.

For more information, see "Solaris Secure Shell and Login Environment Variables" in the *System Administration Guide: Security Services*. See also the `sshd_config`(4) man page.

New Password Options for Nonlogin and Locked Accounts

This feature was introduced in the Solaris Express 10/04 release.

The `passwd` command has two new options, `-N` and `-u`. The `-N` option creates a password entry for a nonlogin account. This option is useful for accounts that should not be logged in to, but must run `cron` jobs. The `-u` option unlocks a previously locked account.

For more information, see the `passwd`(1) man page.

`-setcond` Option to `auditconfig` Command Is Removed

This feature was introduced in the Solaris Express 10/04 release.

The `-setcond` option to the `auditconfig` command has been removed. To temporarily disable auditing, use the `audit -t` command. To restart auditing, use the `audit -s` command.

`perzone` Audit Policy

This feature was introduced in the Solaris Express 8/04 release.

The `perzone` audit policy enables non-global zones to be audited individually. A separate audit daemon runs in each zone. The daemon uses audit configuration files that are specific to the zone. Also, the audit queue is specific to the zone. By default, the policy is off.

For more information, see the `auditd`(1M) and `auditconfig`(1M) man pages.

Kerberos Enhancements

These Kerberos enhancements are included in the Solaris 10 release. Several of the enhancements were introduced in prior Software Express releases.

- Kerberos protocol support is provided in remote applications, such as `ftp`, `rcp`, `rdist`, `rlogin`, `rsh`, and `telnet`. See the man pages for each command or daemon and the `krb5_auth_rules`(5) man page for more information.

- The Kerberos principal database can now be transferred by incremental update instead of transferring the entire database each time. Incremental propagation provides several advantages including the following:

  - Increased database consistencies across servers
  - The need for fewer resources, such as network and CPU resources
  - Much more timely propagation of updates
  - An automated method of propagation

- A new script helps automatically configure a Kerberos client. The script helps an administrator quickly and easily set up a Kerberos client. For procedures that use the new script, see Chapter 22, "Configuring the Kerberos Service (Tasks)," in the *System Administration Guide: Security Services*. See also the `kclient`(1M) man page for more information.

- Several new encryption types have been added to the Kerberos service. These new encryption types increase security and enhance compatibility with other Kerberos implementations that support these encryption types. All of the encryption types are documented in the `mech`(4) man page. See "Using Kerberos Encryption Types" in the *System Administration Guide: Security Services* for more information. The encryption types offer the following capabilities:

  - AES encryption type can be used for high-speed, high-security encryption of Kerberos sessions. The use of AES is enabled through the Cryptographic Framework.

  - ARCFOUR-HMAC provides better compatibility with other Kerberos versions.

  - Triple DES (3DES) with SHA1 increases security. This encryption type also enhances interoperability with other Kerberos implementations that support this encryption type.

- The KDC software and `kinit` command now support the use of the TCP network protocol. This addition provides more robust operation and better interoperability with other Kerberos implementations. The KDC now "listens" on both the traditional UDP ports and on the TCP ports so that it can respond to requests that use either protocol. The `kinit` command first tries UDP when sending a request to the KDC. If a failure occurs, the `kinit` command then tries TCP.

- Support for IPv6 was added to the KDC software with `kinit`, `klist`, and `kprop` commands. Support for IPv6 addresses is provided by default. No configuration parameters need to change to enable this support.

- A new `-e` option has been added to several subcommands of the `kadmin` command. This new option allows for the selection of the encryption type when creating principals. See the `kadmin`(1M) man page for more information.

- Additions to the `pam_krb5` module manage the Kerberos credentials cache by using the PAM framework. See the `pam_krb5`(5) man page for more information.

- Support is provided for auto-discovery of the Kerberos KDC, admin server, `kpasswd` server, and host or domain name-to-realm mappings that use DNS lookups. This support reduces some of the steps that are needed to install a Kerberos client. The client is able to locate a KDC server by using DNS instead of reading a configuration file. See the `krb5.conf`(4) man page for more information.

- A new PAM module called `pam_krb5_migrate` has been introduced. The new module helps in the automatic migration of users to the local Kerberos realm if the users do not already have Kerberos accounts. See the `pam_krb5_migrate`(5) man page for more information.

- The `~/.k5login` file can now be used with GSS applications, `ftp` and `ssh`. For more information, see the `krb5_auth_rules`(5) man page.

- The `kproplog` utility has been updated to display all attribute names per log entry. For more information, see the `kproplog`(1M) man page.

- A new configuration file option makes the strict TGT verification feature optionally configurable on a per-realm basis. See the `krb5.conf`(4) man page for more information.

- Extensions to the password-changing utilities enable the Solaris Kerberos V5 administration server to accept password change requests from non-Solaris clients. See the `kadmin`(1M) man page for more information.

- The default location of the replay cache has been moved from RAM-based file systems to persistent storage in `/var/krb5/rcache/`. The new location protects against replays if a system is rebooted. Performance enhancements were made to the rcache code. However, overall replay cache performance might be slower because of the use of persistent storage.

- The replay cache can now be configured to use file storage or memory-only storage. Refer to the `krb5envvar`(5) man page for more information about environment variables that can be configured for key table and credential cache types or locations.

- The GSS credential table is no longer necessary for the Kerberos GSS mechanism. For more information, see the `gsscred`(1M), `gssd`(1M), and `gsscred.conf`(4) man pages.

- The Kerberos utilities, `kinit` and `ktutil`, are now based on MIT Kerberos version 1.2.1. This change added new options to the `kinit` command and new subcommands to the `ktutil` command. For more information, see the `kinit`(1) and the `ktutil`(1) man pages.

- The Solaris Kerberos Key Distribution Center (KDC) is now based on MIT Kerberos version 1.2.1. The KDC now defaults to a btree-based database, which is more reliable than the current hash-based database. See the `kdb5_util`(1M) man page for more information. For Solaris 9 users, this change was introduced in the Solaris 9 12/03 release.

TCP Wrappers for `rpcbind`

This feature was introduced in the Solaris Express 4/04 release.

TCP wrapper support has been added to the `rpcbind` command. This support allows the administrator to limit calls to `rpcbind` to selected hosts. The administrator can also log all calls to `rpcbind`.

For further information, see the `rpcbind`(1M) man page.

`zonename` Audit Token and Audit Policy Option

The Solaris Zones partitioning technology was introduced in the Solaris Express 2/04 release. See "Solaris Zones Software Partitioning Technology" on page 35. The related `zonename` enhancements that are described here were also introduced in the Solaris Express 2/04 release.

The `zonename` audit token records the name of the zone in which an audit event occurred. The `zonename audit policy` option determines, for all zones, whether the `zonename` token is included in audit records. If the criteria for audit class preselection vary between non-global zones, then you might want to analyze audit records by zone. The `zonename` audit policy enables you to postselect audit records by zone.

See "Auditing and Solaris Zones" in the *System Administration Guide: Security Services*.

For further information, see the `audit.log`(4), `auditconfig`(1M), and `auditreduce`(1M) man pages. See also "Using Solaris Auditing in Zones" in the *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*.

User Commands for Solaris Cryptographic Framework

This feature was introduced in the Solaris Express 1/04 release.

The `digest`, `mac`, and `encrypt` commands now include an option to list the algorithms that are available for each command. For the `mac` and `encrypt` commands, the output includes the key lengths that each algorithm accepts. Also, the `-I <IV-file>` option has been removed from the `encrypt` and `decrypt` commands.

For further information, see Chapter 14, "Solaris Cryptographic Framework (Tasks)" and "Protecting Files With the Solaris Cryptographic Framework" in the *System Administration Guide: Security Services*.

For more information, see the `encrypt`(1), `digest`(1), and `mac`(1) man pages.

IKE Configuration Parameters

This feature was introduced in the Solaris Express 1/04 release.

Retransmission parameters and packet time-out parameters have been added to the `/etc/inet/ike/config` file. The parameters enable the administrator to tune the IKE Phase 1 (Main Mode) negotiation. The tuning enables Solaris IKE to interoperate with platforms that implement the IKE protocol differently. The parameters also help the administrator adjust for network interference and heavy network traffic.

For a detailed description of the parameters, see the `ike.config(4)` man page.

Simple Authentication and Security Layer

This feature was introduced in the Solaris Express 12/03 release.

Simple Authentication and Security Layer (SASL) provides developers of applications with interfaces for adding authentication, data integrity checking, and encryption to connection-based protocols.

For further information, see "Simple Authentication and Security Layer for Developers" on page 96.

See also Chapter 17, "Using SASL," in the *System Administration Guide: Security Services*.

Audit Time Now Reported in ISO 8601 Format

This feature was introduced in the Solaris Express 12/03 release.

The file and header tokens in audit records now report time in ISO 8601 format. For example, the output from the `praudit` command for the file token is as follows:

Old File Token:

```
file,Mon Oct  13 11:21:35 PDT 2003, + 506 msec,
/var/audit/20031013175058.20031013182135.machine1
```

New File Token:

```
file,2003-10-13 11:21:35.506 -07:00,
/var/audit/20031013175058.20031013182135.machine1
```

Old Header Token:

```
header,173,2,settppriv(2),,machine1,
Mon Oct 13 11:23:31 PDT 2003, + 50 msec
```

New Header Token:

```
header,173,2,settppriv(2),,machine1,
2003-10-13 11:23:31.050 -07:00
```

The XML output has also changed. For example, the output from the `praudit -x` command formats the file token as follows:

```
<file iso8601="2003-10-13 11:21:35.506 -07:00">
/var/audit/20031013175058.20031013182135.machine1</file>
```

Customized scripts or tools that parse `praudit` output might need to be updated to accommodate this change.

For further information, see Chapter 27, "Solaris Auditing (Overview)" and "Changes to Solaris Auditing for the Solaris 10 Release" in the *System Administration Guide: Security Services*.

Basic Audit and Reporting Tool

This feature was introduced in the Solaris Express 11/03 release.

Basic Audit and Reporting Tool (BART) is a command-line utility that enables OEMs, advanced users, and system administrators to do a check at file level of the software contents of a target system. The utility is useful for gathering information about what is installed on a system. BART also enables you to compare installed systems, and to compare the contents of a system over time.

For further information, see Chapter 5, "Using the Basic Audit Reporting Tool (Tasks)," in the *System Administration Guide: Security Services*.

See also the `bart_manifest`(4), `bart_rules`(4), and `bart`(1M) man pages.

IPsec and the Solaris Cryptographic Framework

This feature was introduced in the Solaris Express 9/03 release.

IPsec uses the Solaris Cryptographic Framework instead of its own encryption and authentication modules. The modules are optimized for the SPARC platform. In addition, a new `ipsecalgs` command-line utility and APIs are provided to query the list of supported IPsec algorithms and other IPsec properties.

For further information, see the `ipsecalgs`(1M) man page.

In the *System Administration Guide: IP Services*, see Chapter 17, "IP Security Architecture (Overview)" and "Authentication and Encryption Algorithms in IPsec."

Solaris Cryptographic Framework for System Administrators

This feature was introduced in the Software Express pilot program.

The Solaris Cryptographic Framework provides cryptographic services to applications in the Solaris environment. The system administrator controls which encryption algorithms can be used through the `cryptoadm` command. The `cryptoadm` command enables you to perform the following functions:

- Manage available providers of cryptographic services
- Set cryptographic security policy, such as disabling algorithms from a particular provider

The framework comes with plug-ins for the AES, DES/3DES, RC4, MD5, SHA-1, DSA, RSA, and Diffie-Hellman algorithms. Plug-ins can be added or removed as needed.

The `encrypt`, `decrypt`, `digest`, and `mac` commands all use cryptographic algorithms from the framework.

For further information, see Chapter 13, "Solaris Cryptographic Framework (Overview)," in the *System Administration Guide: Security Services*.

See also the following man pages:

- `cryptoadm`(1M)
- `kcfd`(1M)
- `libpkcs11`(3LIB)
- `pkcs11_kernel`(5)
- `pkcs11_softtoken`(5)

Remote Audit Log

This feature was introduced in the Software Express pilot program.

In addition to recording audit events in the binary audit log, the Solaris releases enable you to record audit events to syslog.

The generation of syslog data allows you to use the same management and analysis tools that are available for syslog messages from a variety of Solaris and non-Solaris environments, including workstation, servers, firewalls, and routers. By using `syslog.conf` to route audit messages to remote storage, you protect log data from alteration or deletion by an attacker. However, the syslog option provides only a summary of audit record data. Also, when syslog data is stored on a remote system, the data is susceptible to network attacks such as denial of service and false or "spoofed" source addresses.

For further information, see Chapter 27, "Solaris Auditing (Overview)" and "Audit Files" in the *System Administration Guide: Security Services*.

See also the following man pages:

- `audit`(1M)
- `audit.log`(4)
- `audit_control`(4)
- `audit_syslog`(5)
- `syslog`(3C)
- `syslog.conf`(4)

FTP Server Enhancements

This feature was introduced in the Software Express pilot program.

Scalability and transfer logging enhancements have been made to the FTP server including:

- The `sendfile()` function is used for binary downloads.
- New capabilities are supported in the `ftpaccess` file:
  - `flush-wait` controls the behavior at the end of a download or directory listing.
  - `ipcos` sets the IP Class of Service for either the control or data connection.

- **passive ports** can be configured so that the kernel selects the TCP port to listen on.
- **quota-info** enables retrieval of quota information.
- **recvbuf** sets the receive (upload) buffer size that is used for binary transfers.
- **rhostlookup** allows or disallows the lookup of the remote host's name.
- **sendbuf** sets the send (download) buffer size that is used for binary transfers.
- **xferlog format** customizes the format of the transfer log entry.

- A new `-4` option makes the FTP server only listen for connections on an IPv4 socket when running in standalone mode.

The FTP client and server now support Kerberos. For more information, refer to the `ftp`(4) man page and to "Kerberos User Commands" in the *System Administration Guide: Security Services*.

In addition, `ftpcount` and `ftpwho` now support the `-v` option, which displays user counts and process information for FTP server classes that are defined in virtual host `ftpaccess` files.

For further information about these changes, see the following man pages:

- `in.ftpd`(1M)
- `ftpaccess`(4)
- `ftpcount`(1)
- `ftpwho`(1)
- `sendfile`(3EXT)

FTP Client

This feature was introduced in the Software Express pilot program.

The Solaris software includes a change to the FTP client. By default, a Solaris FTP client, connected to a Solaris FTP server, lists directories as well as plain files when the `ls` command is issued. If the FTP server is not running in the Solaris operating system, directories may not be listed.

To allow for the default Solaris behavior when connecting to non-Solaris FTP servers, the `/etc/default/ftp` file can be edited on each Solaris client. To make the change for individual users, the `FTP_LS_SENDS_NLST` environment variable can be set.

For more information, see the `ftp`(4) man page.

The FTP client and server now support Kerberos. For more information, refer to the `ftp`(4) man page and to "Kerberos User Commands" in the *System Administration Guide: Security Services*.

Internet Key Exchange (IKE) Key Storage on Sun Crypto Accelerator 4000 Board

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/03 release.

IKE now runs on IPv6 as well as IPv4 networks. For information about keywords that are specific to the IPv6 implementation, see the ifconfig(1M) and ike.config(4) man pages.

When a Sun Crypto Accelerator 4000 board is attached, IKE can offload computation-intensive operations to the board, thus freeing the operating system for other tasks. IKE can also use the attached board to store public keys, private keys, and public certificates. Key storage on a separate piece of hardware provides additional security.

For further information, see the ikecert(1M) man page.

See also the following in *System Administration Guide: IP Services*:

- "IP Security Architecture (Overview)"
- "Internet Key Exchange (Overview)"
- "IKE and Hardware Storage"
- "Configuring IKE (Tasks)"
- "Configuring IKE to Find Attached Hardware"

IKE Hardware Acceleration

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

Public-key operations in IKE can be accelerated by a Sun Crypto Accelerator 1000 card and Sun Crypto Accelerator 4000 cards. The operations are offloaded to the card. The offloading accelerates encryption and reduces demands on Solaris Operating System resources.

For information about IKE, see the following in *System Administration Guide: IP Services*:

- "Configuring IKE to Find Attached Hardware"
- "Internet Key Exchange (Overview)"
- "IKE and Hardware Acceleration"
- "Configuring IKE (Tasks)"
- "Configuring IKE to Find Attached Hardware"

ipseckey Enhancement

This feature was introduced in the Software Express pilot program.

For network administrators who have installed IPsec or IKE on their systems, the `ipseckey` parser provides clearer help. The `ipseckey monitor` command now provides a timestamp for each event.

For more information, see the `ipseckey`(1M) man page.

Credential Propagation Over Loopback Connections

This feature was introduced in the Software Express pilot program.

This Solaris release introduces `ucred_t *` as an abstract representation of the credentials of a process. These credentials can be retrieved by using `door_ucred()` in door servers and `getpeerucred()` for loopback connections. The credentials can be received by using `recvmsg()`.

See the `socket.h`(3HEAD) man page for further information.

Auditing Header Token Contains Host Information

This feature was introduced in the Software Express pilot program.

The header token in audit records has been expanded to include the name of the host.

The old header is displayed as follows:

```
header,131,4,login - local,,Wed Dec 11 14:23:54 2002, + 471 msec
```

The new expanded header is displayed as follows:

```
header,162,4,login - local,,example-hostname,
Fri Mar 07 22:27:49 2003, + 770 msec
```

Customized scripts or tools that parse `praudit` output might need to be updated to reflect this change.

See Chapter 30, "Solaris Auditing (Reference)" and "`header` Token" in the *System Administration Guide: Security Services* for further information.

Auditing Enhancements

This feature was introduced in the Software Express pilot program and in the Solaris 9 8/03 release.

Enhancements to the audit features in the Solaris software reduce noise in the trail, and enable administrators to use XML scripting to parse the trail. These enhancements include the following:

- Public files are no longer audited for read-only events. The public policy flag for the auditconfig command controls whether public files are audited. By not auditing public objects, the audit trail is greatly reduced. Attempts to read sensitive files are therefore easier to monitor.

- The praudit command has an additional output format, XML. The XML format enables the output to be read in a browser, and provides source for XML scripting for reports. See the praudit(1M) man page.

- The default set of audit classes has been restructured. Audit metaclasses provide support for finer-grained audit classes. See the audit_class(4) man page.

- The bsmconv command no longer disables the use of the Stop-A key combination. The Stop-A event is now audited to maintain security.

For further information, see the following sections in the *System Administration Guide: Security Services*:

- "Solaris Auditing (Reference)"
- "Definitions of Audit Classes"
- "praudit Command"
- "Solaris Auditing (Overview)"
- "Audit Terminology and Concepts"
- "Changes to Solaris Auditing for the Solaris 10 Release"

New Audit Token path_attr

This feature was introduced in the Software Express pilot program.

The path_attr audit token contains access path information for an object. The access path specifies the sequence of attribute file objects below the path token object. Systems calls such as openat() access attribute files. For more information on extended file attributes, see the fsattr(5) man page.

The path_attr token has three fields:

- A token ID field that identifies this token as a path_attr token
- A count that represents the number of sections of attribute file paths
- One or more null-terminated strings

The praudit command displays the path_attr token as follows:

```
path_attr,1,attr_file_name
```

For further information, see Chapter 30, "Solaris Auditing (Reference)" and "path_attr Token" in the *System Administration Guide: Security Services*.

Password History Checking

This feature was introduced in the Software Express pilot program.

For login accounts that are defined in local files, a password history of up to 26 previously changed passwords can be enabled. When a user changes a password, the attempt fails if the new password matches one of the passwords in the history. Also, login name checking can be disabled.

See the `passwd`(1) man page for more information.

Enhanced `crypt()` Function

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

Password encryption protects passwords from being read by intruders. Three strong password encryption modules are now available in the software:

- A version of Blowfish that is compatible with Berkeley Software Distribution (BSD) systems
- A version of Message Digest 5 (MD5) that is compatible with BSD and Linux systems
- A stronger version of MD5 that is compatible with other Solaris systems

For information on how to protect your user passwords with these new encryption modules, see the following sections in the *System Administration Guide: Security Services*:

- "Controlling Access to Systems (Tasks)"
- "Changing the Default Algorithm for Password Encryption (Task Map)"
- "Managing Machine Security (Overview)"
- "Changing the Default Algorithm for Password Encryption"

For information on the strength of the modules, see the `crypt_bsdbf`(5), `crypt_bsdmd5`(5), and `crypt_sunmd5`(5) man pages.

System Performance Enhancements

This section describes all system performance features in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002. The "New Architecture for Network Stacks" description is of particular importance.

New Architecture for Network Stacks

This new architecture is of key importance in the Solaris 10 system. This feature was introduced in the Solaris Express 10/03 release.

The networking stack for TCP connections has been rearchitected to deliver extremely high performance while improving the scalability. This innovation enables Solaris networking to vertically partition the workload by using an IP classifier-based lockless design. This design reduces the overheads of synchronization and reduces cross-communication between CPUs. The improvements are visible across all networking workloads and are transparent to the users.

CPU Performance Counters

The feature was introduced in the Solaris Express 4/04 release.

The CPU Performance Counter (CPC) system gives better access to the performance analysis features available on SPARC and x86 platforms.

The CPC commands, cpustat and cputrack, have enhanced, command-line syntax for specifying CPU information. In previous versions of the Solaris OS, you were required to specify two counters. The configuration of both commands now enables you to specify only one counter. For simple measurements, you can even omit the counter configuration.

For more information, see the cpustat(1M) man page. For more information about using the cputrack command, see the cputrack(1) man page.

Improved System Performance for Large Numbers of Interfaces

This feature was introduced in the Software Express pilot program.

System performance during and after plumbing large numbers of interfaces, such as virtual area local networks (VLANs) and tunnels, has been improved significantly. However, avoid using large numbers (thousands) of routes with large numbers (thousands) of interfaces.

UFS Logging Performance Enhancements

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

UFS logging enables faster system reboots. Because the file system transactions are already stored, checking the file system is not required if the file system is already consistent.

In addition, the performance of UFS logging improves or exceeds the level of performance of nonlogging file systems in this Solaris release. Results of the Standard Performance Evaluation Corporation system file server (SPECsfs) benchmark show that the performance of NFS-mounted file systems that have logging enabled equals the level of non-UFS logging performance. In some I/O-bound configurations, logging UFS file systems exceed the performance of nonlogging UFS file systems by approximately 25 percent. In other tests, performance of logging UFS file systems is 12 times better than the performance of nonlogging UFS file systems.

For information on how to enable logging on a UFS file system, see the *System Administration Guide: Devices and File Systems* or the mount_ufs(1M) man page.

Memory Placement Optimization (MPO)

This feature was introduced in the Software Express pilot program and in the Solaris 9 9/02 release.

The Solaris dispatcher and the virtual memory subsystem are enhanced to optimize the time that is required for an application to access memory. This enhancement can automatically improve the performance of many applications. This feature is currently implemented to support specific platforms such as the Sun Fire™ 3800–6800, the Sun Fire 12K, and the Sun Fire 15K.

A technical white paper about *Solaris Memory Placement Optimization and Sun Fire Servers* is available at the following Web site:

http://sundoc.central.sun.com/dm/index.jsp

Dynamic Intimate Shared Memory (DISM) Large-Page Support

This feature was introduced in the Software Express pilot program and in the Solaris 9 9/02 release.

Large-page support is now provided for Dynamic Intimate Shared Memory (DISM). This large-page support can increase the performance of applications that are capable of dynamically adjusting shared memory sizes.

For further information on shared memory operations, see shmop(2).

Device Management

This section describes all device management features in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002.

New or Updated Drivers

These driver features were new or revised in the Solaris Express 10/04 release.

bge Gigabit Ethernet Driver for Broadcom bcm57xx

The Solaris Express 10/04 release provided an enhanced bge driver to support Broadcom 5700/5701/5705/5782 gigabit Ethernet chipsets on x86 platforms and on the SPARC platform.

For further information, see the bge(7D) man page.

`asy` Handles IXON/IXOFF in Driver

This feature enables a quick response for software flow control on x86 platforms. This quick response avoids situations where `asy` can stall because an XOFF or XON character is embedded in pending data.

For further information, see the `asy`(7D) man page.

ECP Parallel Port Driver

The Solaris OS on x86 platforms originally used the `lp` driver for parallel print ports. This method lacked ECP mode support. With the ECPP driver added, the Solaris OS on x86 platforms provides a robust parallel port driver with ECP mode support.

For further information, see the `ecpp`(7D) man page.

UHCI Driver Enhancement

An interrupt-out transfer has been added to the new Universal Host Controller Interface (UHCI) driver for x86 platforms. This new feature enables users to transfer urgent data to the devices through Interrupt Out in a high priority.

Other new features provided by the new UHCI driver include the following:

- BULK short transfer
- Port enumeration
- Multiple BULK transfer
- Isoc IN transfer
- BULK IN transfer

In this Solaris 10 release, the UHCI driver is available for the both x86 platforms and for the SPARC platform.

For further information, see the `uhci`(7D) man page.

Adaptec AdvancedRAID Controllers Support

Software support of Adaptec AdvancedRAID controllers has been added to the Solaris 10 OS for x86 platforms. This feature enables you to utilize SCSI RAID disk arrays both as data drives and as boot drives. SCSI CD-ROM, DVD, or tapes are not supported, nor are online configurations.

For further information, see the `aac`(7D) man page.

Dell/LSI SCSI RAID PERC Controllers Support

Software support of Dell/LSI SCSI RAID PERC controllers has been added to the Solaris 10 OS for x86 platforms. This feature enables users to utilize SCSI RAID disk arrays both as data drives and as boot drives. SCSI CD-ROM, DVD, or tapes are not supported, nor are online configurations.

For further information, see the `amr`(7D) man page.

Realtek RTL8139 NIC Support

This Solaris 10 release provides x86 platform support for the Realtek fast Ethernet chipset RTL8139.

For further information, see the `rtls`(7D) man page.

`audio810` Audio Driver

The `audio810` device driver has been added to the Solaris 10 OS for x86 platforms. This driver supports the audio controllers that are embedded in the Intel ICH3, ICH4, ICH5, and AMD 8111 southbridge chips. This driver supports Solaris audio mixer interfaces, and enables users to utilize audio hardware to play back or record sound.

For further information, see the `audio810`(7D) man page.

SATA Controllers Support

Software support of selected SATA controllers was added to Solaris 10 OS for x86 platforms. This feature allows users to utilize Serial ATA Hard Drives both as data drives and as boot drives.

For further information, see the `ata`(7D) man page.

1394 (FireWire) and Mass-Storage Support on x86 Systems

This feature was introduced in the Solaris Express 10/04 release.

In this Solaris release, the 1394 OpenHCI host controller driver has been updated to include support for x86 systems. Previously support for 1394 (FireWire) technology was only available on SPARC systems.

IEEE 1394 is also known by the Apple Computer trademark name, FireWire.

1394 is an industry-standard serial bus that supports data rates of 100 Mbit/sec, 200 Mbit/sec, or 400 Mbit/sec. The bus readily handles data from consumer electronics devices, such as video cameras, because of its high bandwidth and isochronous (on-time) capabilities.

For more information, see the `hci1394`(7D) man page.

In this Solaris release, the `scsa1394` driver has been introduced to support 1394 mass-storage devices that are compliant with the Serial Bus Protocol 2 (SBP-2) specification. This driver supports both bus-powered and self-powered 1394 mass storage devices. Previously, only 1394 video cameras were supported.

1394 mass storage devices are treated as removable media devices. A 1394 mass storage device can be formatted by using the `rmformat` command. Using a 1394 mass-storage device is no different than using a USB mass-storage device. You can mount, eject, hot-remove, and hot-insert a 1394 mass-storage device.

For more information about using these devices, see the `hci1394`(7D) man page. See also Chapter 8, "Using USB Devices (Tasks)," in the *System Administration Guide: Devices and File Systems*.

Internet Printing Protocol Listener

This feature was introduced in the Solaris Express 10/04 release.

The Internet Printing Protocol (IPP) listener feature for printing on a network works in conjunction with the Apache web server that is shipped as part of the Solaris software. IPP uses HTTP to transport requests. Using the Apache web server, the listener loads IPP print requests and processes them. The IPP listener listens on port 631 for HTTP requests and then communicates the requests to the printing system.

For more information, see the *System Administration Guide: Advanced Administration*.

Fibre Channel Connectivity for Storage Devices

This feature was introduced in the Solaris Express 8/04 release.

The Solaris Fibre Channel I/O stack provides connectivity and multipathing functions for Fibre Channel devices in a Storage Area Network (SAN). The following enhancements have been made to this feature:

- Boot support with Solaris installation programs on SPARC platforms for Fibre Channel devices
- x86 platform support (32-bit and 64-bit) for Fibre Channel devices

Note – Previous SAN releases were distributed through SunSolve and Sun Download Center (SDLC) in patch sets and packages for the Solaris 8 OS and the Solaris 9 OS. All new SAN features will be integrated and distributed through Solaris releases.

For more information about SAN and related documents, refer to `http://www.sun.com/storage/san/`.

Expanded Printer Support

This feature was introduced in the Solaris Express 6/04 release and in the Solaris 9 9/04 release.

In this release, modifications have been made to incorporate support for a wide array of printers. This support is accomplished through the use of additional transformation software, raster image processor (RIP), and PostScript™ Printer Description (PPD) files.

These additions provide functionality that enables you to print to printers, such as the Lexmark Optra E312 and Epson Stylus Photo 1280, by using PPD files.

This feature is useful in an environment where printers do not have resident PostScript processing capabilities.

In addition, the existing Solaris printing tools have been modified to include a new -n option to the lpadmin command. With this option, you can designate a PPD file to use when creating a new print queue or when modifying an existing print queue.

Also, the Solaris Print Manager screens have been updated to enable you to choose a PPD file for the print queue through the selection of make, model, and driver. This new feature differs greatly from previous Solaris software releases. In previous releases, the provided list of printer types, and information about whether the printer accepted PostScript or ASCII text, was limited.

For more information, see the lpadmin(1M) man page and the *System Administration Guide: Advanced Administration*.

Common Solaris Target Disk Driver

This feature was introduced in the Solaris Express 5/04 release.

In this Solaris release, the disk drivers for the SPARC and the x86 platforms are merged into a single driver. This change creates one source file for the following 3 drivers:

- SPARC sd for SCSI devices
- x86 sd for Fibre Channel and SCSI devices
- SPARC ssd for Fibre Channel devices

In previous Solaris releases, three separate drivers were needed to provide support for SCSI and Fibre Channel disk devices on the SPARC and x86 platforms. All of the disk utilities, such as the format, fmthard, and fdisk commands, have been updated to support these changes.

For more information, see the sd(7D) and ssd(7D) man pages.

In addition, Solaris support for the EFI disk label is now available on x86 systems. For further information about EFI disk labels, see .

For further information, see also "What's New in Disk Management in the Solaris 10 Release?" in the *System Administration Guide: Devices and File Systems*.

Wheel Mouse Support

Support for the following mouse features was added in the Solaris Express 4/04 release and in the Solaris 9 9/04 release:

- Wheel mouse scrolling is available on a USB mouse device

  This support means that rolling the wheel on a USB or a PS/2 mouse results in a "scroll" in the application or window under mouse focus.

  StarOffice and Mozilla applications support wheel mouse scrolling. However, other applications might not support wheel mouse scrolling.

- Support for more than 3 buttons on USB or PS/2 mouse devices

USB 2.0 Features

This description was introduced in the Solaris Express 2/03 release and updated in the Solaris Express 9/03 release. For Solaris 9 users, USB 2.0 support was introduced in the Solaris 9 4/04 release.

The Software Express releases include the following USB 2.0 features:

- **Better Performance** – Increased data throughput for devices attached to USB 2.0 controllers, up to 40 times faster than USB 1.1 devices

  You can particularly benefit from the high-speed USB protocol when accessing high-speed mass storage devices, such as DVDs and hard drives.

- **Backward Compatibility** – Compatibility with 1.0 and 1.1 devices and drivers so that you can use the same connectors and software interfaces

  USB 1.x devices can also use the same 1.x cables as before or the newer USB 2.0 cables. USB 2.0 devices, per next description, require USB 2.0 cables.

For a description of USB devices and terminology, see "Overview of USB Devices" in the *System Administration Guide: Devices and File Systems*.

USB 2.0 Devices

This description was introduced in the Solaris Express 2/03 release and updated in the Solaris Express 9/03 release. For Solaris 9 users, USB 2.0 device support was introduced in the Solaris 9 4/04 release.

USB 2.0 devices are defined as high-speed devices that follow the USB 2.0 specification. You can refer to the USB 2.0 specification at `http://www.usb.org`.

Some of the USB devices that are supported in this Solaris release are as follows:

- Mass storage devices – CD-RWs, hard disks, DVD, digital cameras, Zip drives, diskettes, and tape drives
- Keyboard, mouse devices, speakers, and microphones
- Audio devices

For a full listing of USB devices that have been verified on the Solaris release, go to:

`http://www.sun.com/io_technologies/ihvindex.html`

Additional storage devices might work by modifying the `scsa2usb.conf` file. For more information, see the `scsa2usb`(7D) man page.

Solaris USB 2.0 device support includes the following features:

- Increased USB bus speed from 12 Mbps to 480 Mbps. This increase means devices that support the USB 2.0 specification can run significantly faster than their USB 1.1 counterparts when they are connected to a USB 2.0 port.

  A USB 2.0 port is defined on SPARC and x86 systems as follows:

  - A port on a USB 2.0 PCI card
  - A port on a USB 2.0 hub that is connected to a USB 2.0 port
  - A system USB port, assuming that it has a USB 2.0 port on the motherboard

- USB 2.0 is Solaris Ready on all PCI-based platforms. A USB 2.0 PCI card is needed to provide USB 2.0 ports. For a list of USB 2.0 PCI cards that have been verified for the Solaris release, go to
  `http://www.sun.com/io_technologies/ihvindex.html`.

- USB 1.1 devices work as they have in the past, even if you have both USB 1.1 and USB 2.0 devices on the same system.

- While USB 2.0 devices operate on a USB 1.x port, their performance is significantly better when connected to a USB 2.0 port.

For more information on USB 2.0 device support, see the `ehci`(7D) and `usba`(7D) man pages.

For information about USB cables and bus-powered devices, see "About USB in the Solaris OS" in the *System Administration Guide: Devices and File Systems*.

Solaris Support for USB Devices

This description was introduced in the Solaris Express 9/03 release.

USB 2.0 support for x86 systems was first introduced in the Solaris 9 4/04 release.

Support of USB 2.0 devices in the Solaris 8 and Solaris 9 releases is provided through the USBA 1.0 version of the USB Dual Framework. The USBA 2.0 framework in the Solaris 10 release supercedes the USBA 1.0 version.

USB Mass Storage Devices

This description was introduced in the Solaris Express 2/03 release and updated in the Solaris Express 9/03 release. For Solaris 9 users, this feature was introduced in the Solaris 9 4/04 release.

All USB storage devices in the Software Express releases are now accessed as removable media devices through volume management. This change has the following advantages:

- USB storage devices with standard MS-DOS or Windows (FAT) file systems are now supported.

- You can use the user-friendly `rmformat` command instead of the `format` command to format and partition all USB storage devices. If you need the functionality of the `format` command, use the `format -e` command.

- You can use the `fdisk` command if you need to do `fdisk`-style partitioning.

- Nonroot users can now access USB storage devices, because the root-privileged `mount` command is no longer needed. The device is automatically mounted by `vold` and is available under the `/rmdisk` directory. If a new device is connected while the system is down, do a reconfiguration boot with the `boot -r` command so that `vold` recognizes the device. If a new device is connected while the system is up, restart `vold`. For more information, refer to the `vold`(1M) and `scsa2usb`(7D) man pages.

- Disks with FAT file systems can be mounted and accessed. See the following example.

  ```
  mount -F pcfs /dev/dsk/c2t0d0s0:c /mnt
  ```

- All USB storage devices are now power managed, except for those that support `LOG SENSE` pages. Devices with `LOG SENSE` pages are usually SCSI drives that are connected through a USB-to-SCSI bridge device. In previous Solaris releases, some USB storage devices were not power managed because they were not recognized as removable media.

- Applications might work differently with USB mass storage devices.

  Note the following issues when using applications with USB storage devices.

  - Applications might make incorrect assumptions about the size of the media because only smaller devices such as diskettes and Zip drives were removable previously.

  - Requests by applications to eject media on devices where this would be inapplicable, such as a hard drive, succeed and do nothing.

  To revert to the behavior of previous Solaris releases that did not treat all USB mass storage as removable media devices, update the `/kernel/drv/scsa2usb.conf` file.

For more information about using USB mass storage devices, see the `scsa2usb`(7D) man page.

For information on troubleshooting USB mass storage device problems, see "What's New in USB Devices?" in the *System Administration Guide: Devices and File Systems*.

USB Driver Enhancements

This description was introduced in the Solaris Express 2/03 release and updated in the Solaris Express 9/03 release. For Solaris 9 users, this feature was introduced in the Solaris 9 4/04 release.

This section describes USB driver enhancements in the Software Express releases.

- **New generic USB driver –** USB devices can now be accessed and manipulated by applications that use standard UNIX `read(2)` and `write(2)` system calls, and without writing a special kernel driver.

    Additional features include the following:

    - Gives applications access to raw device data and device status
    - Supports control, bulk, and interrupt (in and out) transfers

    For more information, refer to the `ugen(7D)` man page and the USB Driver Development Kit at `http://developers.sun.com/`.

- **Digi Edgeport USB support –** Provides support for several Digi Edgeport USB-to-serial-port converter devices.

    - New devices are accessed as `/dev/term/[0-9]*` and `/dev/cua/[0-9]*`.
    - USB serial ports are usable as any other serial port would be, except that they cannot serve as a local serial console. The fact that their data is run through a USB port is transparent to the user.

    For more information, see the `usbser_edge(7D)` man page, or go to `http://www.digi.com` and `http://www.sun.com/io`.

- **Documentation and binary support for user-written kernel and userland drivers – A** Solaris 9 USB Driver Development Kit (DDK) is available, and its documentation is applicable to the Solaris 10 release. For up-to-date information about USB driver development, including information on the Driver Development Kit, go to `http://developers.sun.com/`.

`EHCI` and `OHCI` Drivers

This description was introduced in the Solaris Express 9/03 release. For Solaris 9 users, this feature was introduced in the Solaris 9 4/04 release.

Features of the `EHCI` driver include the following:

- Compliance with enhanced host controller interface that supports USB 2.0.
- Support for high-speed control, bulk, and interrupt transfers.
- Currently, no support is available for high-speed isochronous transactions. For example, you cannot connect USB 1.x devices to a 2.0 hub that is connected to a USB 2.0 port.

Note the following when using EHCI and OHCI controlled devices:

- A USB 2.0 host controller has one high-speed Enhanced Host Controller (EHCI) and one or more low-speed or full-speed OpenHCI Host Controller (OHCI) embedded controllers. Devices connected to a USB 2.0 port are dynamically assigned to either an EHCI or OHCI controller, depending on whether they support USB 2.0.

- If USB 2.0 and USB 1.x devices are on the system, the EHCI and OHCI drivers "hand-off" device control, depending on the type of device that is connected to the system.

Note – USB 2.0 storage devices connected to a port on a USB 2.0 PCI card, and that were used with a prior Solaris release in the same hardware configuration, can change device names after upgrading to this release. This change occurs because these devices are now seen as USB 2.0 devices and are taken over by the EHCI controller. The controller number, w in /dev/[r]dsk/cwtxdysz, is changed for these devices.

For further information, see the man pages ehci(7D) and ohci(7D).

Logical Unit Reset

This feature was introduced in the Software Express pilot program.

The Solaris SCSI disk driver now supports SCSI logical unit reset. This feature improves error recovery processing on multi-LUN SCSI target devices. This feature can reset a logical unit without affecting other logical units on that device.

The ability to reset LUNs is particularly useful in conjunction with multiple pathing and with storage units that provide a large number of LUNs. For more information, see the scsi_reset(9F) and the tran_reset(9E) man pages.

Networking Enhancements

This section describes all networking enhancements in the Solaris 10 release that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002.

Virtual IP Source Address Selection

This feature was introduced in the Solaris 10 release.

Virtual IP source address selection enables a system administrator to specify an IP source address to be used for packets that are routed through a particular network interface. This source address can be hosted on the just-introduced virtual network interface (vni), which is immune to hardware failures. Alternatively, the source address can be hosted on the loopback interface.

You can use virtual IP source address selection in conjunction with IP routing protocols to provide multipathing (that is, redundancy) at the network layer, beyond the first router. Currently, this form of multipathing works in conjunction with the RIPv2 routing protocol that is available in the `in.routed` daemon.

For more information about the virtual IP source address selection feature, see the `ifconfig`(1M) and `vni`(7d) man pages.

Stream Control Transmission Protocol

This feature was introduced in the Solaris Express 8/04 release.

Stream Control Transmission Protocol (SCTP) is a reliable transport protocol that is now included in the Solaris Operating System's TCP/IP protocol stack. SCTP provides services that are similar to TCP. However, SCTP supports connections between endpoints that are multihomed, that is, with more than one IP address. The support for multihoming makes SCTP a popular transport protocol for telephony applications. SCTP also supports multistreaming and partial reliability.

The SCTP protocol does not require additional configuration after the Solaris 10 OS is installed. However, you might need to add service definitions, so that particular applications can run over SCTP.

For information about configuring SCTP, refer to the *System Administration Guide: IP Services*.

Zebra Multiprotocol Routing Suite

This feature was introduced in the Solaris Express 8/04 release.

The Solaris 10 OS now includes the Zebra multiprotocol routing suite. This suite includes Open Source Zebra 0.92a routing software with bug fixes for Sun platforms. Now system administrators can use the well-known open-source routing protocols RIP, BGP, and OSPF for administering their Solaris based networks. Moreover, the OSPF daemon can be used for high network availability on multihomed servers. The Zebra packages contain these protocols and the `zebraadm` administration tool.

Refer to the `/etc/sfw/zebra/README.Solaris` file for configuration information and other details.

IPsec and NAT Traversal

This feature was introduced in the Solaris Express 8/04 release.

IKE can now initiate IPsec security associations from behind a Network Address Translation (NAT) box. Only the ESP protocol over an IPv4 network is allowed. Additionally, IPsec security associations for traffic that traverses a NAT cannot be accelerated with the Solaris Crypto Accelerator 4000 board. IKE acceleration is unaffected.

For more information, see the ipseckey(1M) man page.

Enhancement to the `nfsmapid` Daemon

This feature was introduced in the Solaris Express 8/04 release and updated in the Solaris 10 release.

In NFS version 4, the nfsmapid daemon provides a mapping from a numeric user identification (UID) or a numeric group identification (GID) to a string representation, as well as the reverse. The string representation is used by the NFS version 4 protocol to represent owner or owner_group.

For example, the UID 123456 for the user, known_user, that is operating on a client that is named system.anydomain.com, would be mapped to known_user@anydomain.com. The NFS client sends the string representation, known_user@anydomain.com, to the NFS server. The NFS server maps the string representation, known_user@anydomain.com, to the unique UID 123456. nfsmapid uses the passwd and group entries in the /etc/nsswitch.conf file to determine which database will be consulted to perform the mappings.

For nfsmapid to work properly, clients and servers on NFS version 4 must have the same domain. To ensure that clients and servers have the same domain, nfsmapid configures the domain by following these strict precedence rules:

- The daemon first checks the /etc/default/nfs file for a value that has been assigned to the NFSMAPID_DOMAIN keyword. If a value is found, the assigned value takes precedence over any other settings. The assigned value is appended to the outbound attribute strings and is compared against inbound attribute strings.

- If no value has been assigned to NFSMAPID_DOMAIN, then the daemon checks for a domain name from a DNS TXT record on a DNS name server. To find a specific DNS record, nfsmapid relies on the resolv.conf(4) configuration file.

- If no DNS TXT record provides a domain name, then by default the nfsmapid daemon uses the local DNS domain.

Note – The use of DNS TXT records is preferred. Configure the _nfsv4idmapdomain TXT record on DNS servers that provide domains for NFS version 4 clients and servers. TXT records provide better support for scaling issues and provide a single point of control.

If your network includes multiple DNS domains, but has only a single UID and GID namespace, all clients must use one value for NFSMAPID_DOMAIN. For sites that use DNS, nfsmapid resolves this issue by obtaining the domain name from the value that you assigned to _nfsv4idmapdomain. If your network is not configured to use DNS, during the first system boot the Solaris OS uses the sysidconfig(1M) utility to provide prompts for an NFS version 4 domain name.

For more information, see the nfsmapid(1M) and sysidtool(1M) man pages. See also the *System Administration Guide: Network Services*.

`sendmail` **Version 8.13**

Introduced in the Solaris Express 8/04 release, sendmail version 8.13 is the default in the Solaris 10 OS. Although this new version of sendmail provides many new features, the FallBackSmartHost option is the most significant addition.

Because of the FallBackSmartHost option you no longer need to use main.cf and subsidiary.cf. The main.cf file was used in environments that supported MX records. The subsidiary.cf file was used in environments without a fully operative DNS. In such environments a smart host was used instead of MX records.

The FallBackSmartHost option provides unified configuration. This option operates like an MX record of last possible preference for all environments. To ensure that mail gets delivered to clients, this option, if enabled, provides a well-connected, or "smart," host that serves as a backup or failover for MX records that fail.

sendmail Version 8.13 also provides the following:

- Additional command-line options
- Additional and revised configuration file options
- Additional and revised FEATURE declarations

For more information, see the *System Administration Guide: Network Services*.

`sendmail` **Version 8.12 Uses TCP Wrappers**

This feature was introduced in the Solaris Express 9/03 release.

Note – In the Solaris Express 8/04 release, sendmail version 8.13 is the default. See

TCP wrappers provide a way of implementing access controls by checking the address of a host that is requesting a particular network service against an access control list. Requests are granted or denied, accordingly. Besides providing this access control mechanism, TCP wrappers also log host requests for network services, which is a useful monitoring function. Examples of network services that might be placed under access control include `rlogind`, `telnetd`, and `ftpd`.

In this Solaris release, version 8.12 of `sendmail` now enables the use of TCP wrappers. This check does not bypass other security measures. By enabling TCP wrappers in `sendmail`, a check has been added to validate the source of a network request before the request is granted. See the `hosts_access`(4) man page.

Note – The Solaris 9 release added support for TCP wrappers in `inetd`(1M) and `sshd`(1M).

Sun Java System Message Queue

The Solaris Express 6/04 release introduced Sun Java System Message Queue 3.5 SP1 Platform Edition. This version replaces Sun Java System Message Queue (MQ) 3.0.1, which was previously introduced in the Software Express pilot program.

Note – For Solaris 9 users, the Message Queue 3.0.1 was introduced for the SPARC platform in the Solaris 9 12/02 release. In the Solaris 9 8/03 release, this feature was available for the x86 platform.

Sun Java System Message Queue 3.5 SP1 Platform Edition is an affordable, standards-based, high-performance Messaging System that integrates disparate IT systems. This Message Queue (MQ) is Java Messaging Services (JMS) 1.1 compliant and supports web services messaging through JAXM (SOAP 1.1 with Attachments).

The new 3.5 release, as compared to the previous 3.0.1 release, includes the following new features:

- **C Messaging API** – Native-to-the-wire C-API for connectivity to legacy C/C++ applications. TCP and SSL transports are supported.

- **Client Connection Failover** – On loss of connection, client automatically reconnects to a different broker in the cluster.

- **Advanced Remote Monitoring Capabilities** – JMS-based API enables monitoring of broker statistics, destination statistics, and VM statistics.

- **Support for Sun Cluster in Java Enterprise System** – Sun Cluster Agent for MQ, available in the Java Enterprise System, uses a file-based datastore with high availability (HA) to provide HA functionality to MQ.

- **J2EE 1.4 compatibility** – J2EE Connector Architecture 1.5 support enables MQ 3.5 to be connected to any J2EE 1.4 compliant Application Server through the MQ Resource Adapter.

- **Dynamic Message Flow Control** – Flow control enables management of throughput and load in the System. Control options include Reject Oldest, Reject Newest, Reject Low Priority, and Flow Control.

- **Local Destinations and Cluster Delivery Policies** – Policy configuration enables throughput optimization for message delivery to brokers in a cluster.

For new feature details, refer to the *Sun Java System Message Queue Release Notes* at `http://docs.sun.com`.

Sun Java System Application Server

Sun Java System Application Server Platform Edition 8 was introduced in the Solaris Express 6/04 release.

Previously, Sun Java System Application Server 7 was introduced in the Software Express pilot program for SPARC platforms, and in the Solaris Express 9/03 release for x86 platforms. Sun Java System Application Server Platform Edition 8 replaces Edition 7 in the Solaris Express 6/04 release.

Note – For Solaris 9 users, version 7 of the Application Server was introduced for SPARC platforms in the Solaris 9 12/02 release, and for x86 platforms in the Solaris 9 12/03 release.

Sun Java System Application Server Platform Edition 8 is the J2EE 1.4 compatible application container from Sun Microsystems that is designed for developers and departmental deployments.

This edition provides the following features:

- **J2EE 1.4 compatible** – Supports the latest J2EE technology standards and ensures application portability.

- **Integrated support for JavaServer Faces** – Enables developers to use the latest presentation layer technology, JavaServer™ Faces.

- **Administration GUI and Log Viewer** – Provides improved display and search mechanism for the Log Viewer, thus enhancing administrator productivity.

- **Deploytool** – A GUI-based tool supplements text editors, enabling developers to assemble J2EE components and deploy J2EE applications.

- **Improves Developer productivity** – Improvements made specifically to increase developer productivity include a reduced process count, a reduced memory footprint, an increased server startup speed, and an increased deployment speed.

For important information about this product, see the Sun Java System Application Server product information at `http://www.sun.com/software/`. For new feature details, refer to the *Sun Java System Application Server Platform Edition 8 Release Notes* at `http://docs.sun.com/`.

Using CacheFS With NFS version 4

This feature was introduced in the Solaris Express 6/04 release.

The cache file system, CacheFS™, is a generic, nonvolatile caching mechanism. CacheFS improves the performance of certain file systems by utilizing a small, fast local disk. You can improve the performance of the NFS environment by using CacheFS.

CacheFS works differently with different versions of NFS. For example, if both the client and the back file system are running NFS version 2 or version 3, the files are cached in the front file system for access by the client. However, if both the client and the server are running NFS version 4, the functionality is as follows. When the client makes the initial request to access a file from a CacheFS file system, the request bypasses the front (or cached) file system and goes directly to the back file system. With NFS version 4, files are no longer cached in a front file system. All file access is provided by the back file system. Also, because no files are being cached in the front file system, CacheFS-specific mount options, which are meant to affect the front file system, are ignored. CacheFS-specific mount options do not apply to the back file system.

Note – The first time you configure your system for NFS version 4, a warning appears on the console to indicate that caching is no longer performed.

For further information about NFS version 4, see *System Administration Guide: Network Services*.

Enhancement for `vacation` Utility

This feature was introduced in the Solaris Express 5/04 release.

The `vacation` utility has been enhanced to enable a user to specify which incoming messages receive autogenerated replies. With this enhancement, the user can avoid sharing confidential or contact information with unknown people. Messages from "spammers" or unknown people would not receive a reply.

This enhancement works by matching an incoming sender's email address to a list of domains or email addresses in a `.vacation.filter` file. This file is created by the user and is located in the user's home directory. If a domain or address match is found, a reply is sent. If no match is found, no reply is sent.

For more details, see the `vacation`(1) man page.

`MILTER, sendmail`'s New Mail Filter API

The feature was introduced in the Solaris Express 4/04 release.

`MILTER, sendmail`'s new mail filter API, permits third-party programs to access mail messages as they are being processed in order to filter meta-information and content. This functionality, introduced in the Solaris 10 release, requires the following:

- `sendmail` binary to be compiled with `-DMILTER`, which has been available since the Solaris 9 release

- The file `/usr/lib/libmilter.so`, which is in the Solaris 10 release, to be available

- These files, `/usr/include/libmilter/mfapi.h` and `/usr/include/libmilter/mfdef.h`, to be available

    Both these files are included in the Solaris 10 release.

Thus, with the Solaris 10 release, the user can build the filter and configure `sendmail` to use it.

For further information about `sendmail`, see *System Administration Guide: Network Services*.

IPv6 Advanced Sockets API

This feature was introduced in the Solaris Express 1/04 release.

The IPv6 Advanced Sockets API updates the Solaris Sockets API to meet the current version of RFC 2292. The advanced API provides the functionality needed to manipulate ICMP packets, obtain interface information, and manipulate IPv6 headers.

For further information, see the *Programming Interfaces Guide*.

Contents of `/usr/lib/mail` Have Moved to `/etc/mail/cf`

This feature was introduced in the Solaris Express 1/04 release.

The contents of the `/usr/lib/mail` directory, which might be in a read-only file system, are now in the `/etc/mail/cf` directory, which is writable. This change better supports `m4` configuration. Note, however, these exceptions. The shell scripts `/usr/lib/mail/sh/check-hostname` and `/usr/lib/mail/sh/check-permissions` are now in the `/usr/sbin` directory. For backward compatibility, symbolic links point to each file's new location.

For further information, see the *System Administration Guide: Network Services*.

Added IPv6 Functionality During Solaris Installation

This feature was introduced in the Solaris Express 11/03 release.

Several new IPv6 functions are started when you select Enable IPv6 during Solaris installation.

- The /etc/nsswitch.conf file policies for the hosts database and ipnodes repositories are synchronized so that the same naming repositories are searched for ipnodes and hosts. Now, hosts can resolve any IPv6 addresses that might be in any of the ipnodes repositories.

- Destination address selection has been modified to avoid using an IPv6 address for a remote host if no IPv6 routes serve that host. Instead, an IPv4 address is used, to avoid any delays when connecting to remote hosts.

  For example, consider the case of an IPv6-enabled host on a network with no IPv6 router. Without the presence of a router, a host has no knowledge of IPv6 routes beyond the local link. Previously, the host experienced timeouts when trying to connect to a preferred IPv6 address. With the new feature for destination address selection, the host prefers an IPv4 destination address. This feature eliminates time-out problems.

For further information, see the *System Administration Guide: IP Services*.

IPv6 Temporary Addresses

The feature was introduced in the Solaris Express 11/03 release.

By default, the interface ID of an IPv6 address is autoconfigured with the interface's hardware-specific address. For example, if the interface is an Ethernet card, its interface ID is autoconfigured from the interface's MAC address. However, some system administrators might need to keep private the hardware-specific addresses of one or more interfaces on a node.

The IPv6 Temporary Addresses feature implements the privacy extensions standard that is defined in RFC 3041, "Privacy Extensions for Stateless Autoconfiguration in IPv6." The temporary address feature enables administrators to assign randomly generated, modified EUI-64 format interface IDs to interfaces of an IPv6 node. In addition, the administrators can specify a time limit for the lifetime of a temporary address. After configuration, the IPv6 daemon in.ndpd automatically generates the temporary interface ID for the interface, in addition to the automatically generated, MAC address-based interface ID.

For further information, see the *System Administration Guide: IP Services*.

routeadm Command

This feature was introduced in the Solaris Express 9/03 release.

The new routeadm command enables system administrators to configure IP forwarding and routing on all interfaces of a system. Any settings that are established through routeadm override system defaults that are read from configuration files at boot time.

The `routeadm` command provides options for enabling or disabling the global packet-forwarding function on all IPv4 or IPv6 interfaces of a system. `routeadm` can also be used to set up a system as a router, by turning on routing daemons for all system interfaces. In an IPv6 environment, `routeadm` can also be used to enable or disable dynamic routing for a host.

For further information, see the `routeadm`(1M) man page and the *System Administration Guide: IP Services*.

TCP Multidata Transmit

Multidata transmit (MDT) is available only for systems that run a 64-bit kernel. This feature was introduced in the Software Express pilot program and in Solaris 9 8/03 release.

Multidata transmit enables the network stack to send more than one packet at one time to the network device driver during transmission. Use of this feature reduces the per-packet processing costs by improving the host CPU utilization or network throughput.

The MDT feature is only effective for device drivers that support this feature.

MDT is enabled by default. MDT can be disabled by including the following line in the `/etc/system` file:

```
# ndd -set /dev/ip ip_multidata_outbound 0
```

For further information, see the `ip`(7P) and the `ndd`(1M) man pages.

See also the *STREAMS Programming Guide*.

`ifconfig router` Option

This feature was introduced in the Software Express pilot program.

The new router option of `ifconfig` allows you to configure IP packet forwarding on individual interfaces. `ifconfig router` and `ifconfig -router` enable or disable IP packet forwarding, respectively, for both IPv4 and IPv6 interfaces. The router option sets the IFF_ROUTER interface flag.

These new options replace the ndd variables *interface-name*:`ip_forwarding` and *interface-name*:`ip6_forwarding` for configuring IP packet forwarding on individual interfaces. Though now obsolete, the ndd variables remain in this Solaris release for backward compatibility. You can still use `ip_forwarding` and `ip6_forwarding` without the *interface-name* prefix to configure IP forwarding for all interfaces on a system.

For detailed information, refer to the `ifconfig`(1M), `ip`(7P), and `ip6`(7P) man pages.

Internet Protocol Version 6 (IPv6) Default Address Selection

This feature was introduced in the Software Express pilot program.

The Solaris 10 Operating System provides a documented and deterministic algorithm for doing IPv6 default source and destination address selection. This feature gives system administrators the limited ability to change address selection precedence through use of a policy table.

The IPv6 default address selection feature is a standardized method for IPv6 source and destination address selection.

The selection mechanisms can be configured by using a policy table. For example, you can edit the policy table to give higher precedence to a particular address prefix. Thereafter, addresses that are within the prefix are sorted before other addresses by name look-up APIs. You can also assign labels to source and destination prefixes in the policy table. This assignment makes sure that particular source addresses are only used with particular destination addresses.

To implement IPv6 default address selection, the Solaris Operating System now includes the `/etc/inet/ipaddrsel.conf` file and the `/usr/sbin/ipaddrsel` command. You use `ipaddrsel.conf` to edit the IPv6 default address policy table. Then you use `ipaddrsel` to commit the changes to the policy table.

Additionally, the `ifconfig` command now includes the "preferred" option. This option lets you designate a particular address to be used as the source address for all IPv6 communication.

For detailed information, refer to the `ipaddrsel.conf`(4), `ipaddrsel`(1M), and the `ifconfig`(1M) man pages.

For further information, see the *System Administration Guide: IP Services*.

Disable NFS and automount Services

This feature was introduced in the Software Express pilot program.

The following NFS daemons are typically started by default at boot time by the `rc` scripts: `nfsd`, `mountd`, `statd`, `lockd`, and the automount daemon, `automountd`. Now, if a machine does not require NFS and automount services, the scripts do not start the NFS daemons and the automount daemon.

The following describes the new behavior:

- The `automount` utility, which is called by `/etc/init.d/autofs`, now starts `automountd` at boot time only if the automount maps have a valid entry.

- `/etc/init.d/nfs.server` starts `mountd`, `nfsd`, `statd`, and `lockd` at boot time only if the machine has any NFS exports.

- `/etc/init.d/nfs.client` starts `statd` and `lockd` only if `/etc/vfstab` includes NFS file systems.

NFS and automount services might not be started at boot time. The following commands can start these services when required by a machine:

- The `automount` utility can start the `automountd` daemon.
- The `mount` command with the `-F nfs` option can start the `lockd` and `statd` daemons. The `automountd` daemon also can start `lockd` and `statd`.
- The `share` command with the `-F nfs` option can start the `nfsd`, `mountd`, `lockd`, and `statd` daemons.

This mechanism for disabling NFS and automount services provides the following benefits:

- Extra security that comes from not running unnecessary daemons on a machine.
- A simplified process for exporting file systems. The `nfsd` and `mountd` daemons, and, if necessary, `lockd` and `statd`, are started by the `share` command with the `-nfs` option. So, you no longer have to edit the `/etc/dfs/dfstab` file and then invoke the `/etc/init.d/nfs.server` program. This new behavior permits an NFS export to be configured with a single command, without editing any configuration files. However, if the system reboots, such exports are not resumed automatically, unless the exports are included in the `/etc/dfs/dfstab` file.

For more information, refer to the man pages for mountd(1M), lockd(1M), statd(1M), and nfsd(1M).

For further information, see also the *System Administration Guide: Network Services*.

Internet Protocol Version 6 (IPv6) 6to4 Router

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

IPv6 networks can now transfer packets over Internet Protocol Version 4 (IPv4) networks by configuring one or more routers to support a 6to4 tunnel. System administrators can use 6to4 tunnels as a transitional method for migrating their networks from IPv4 to IPv6. This feature implements RFCs 3056 and 3068.

For further information on IPv6, see the *System Administration Guide: IP Services*.

Packet Tunneling Over IPv6

This feature was introduced in the Software Express pilot program and in the Solaris 9 9/02 release

This feature enables tunneling over IPv6 for both IPv4 over IPv6 tunnels and IPv6 over IPv6 tunnels. IPv4 packets or IPv6 packets can be encapsulated in IPv6 packets.

For further information, see the *System Administration Guide: IP Services*.

Hosting Multiple Web Sites on a Single Solaris Machine

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

The Solaris Network Cache and Accelerator (NCA) kernel module now supports multiple instances of a web server. This support enables you to use a Solaris machine to perform Internet protocol (IP) address-based virtual web hosting. The Solaris software uses a single configuration file, /etc/nca/ncaport.conf, to map NCA sockets to IP addresses.

For further information, see the ncaport.conf(4) man page.

IP Quality of Service

This feature was introduced in the Software Express pilot program and in the Solaris 9 9/02 release.

IP Quality of Service (IPQoS) is a new feature in the Solaris Operating System. IPQoS enables system administrators to provide different levels of network service to customers and to critical applications. By using IPQoS, the administrator can set up service-level agreements. These agreements provide an Internet service provider's (ISP) clients with varying levels of service that are based on a price structure. A company could also use IPQoS to prioritize among applications so that critical applications get a higher quality of service than less critical applications.

For further information, see the *System Administration Guide: IP Services*.

User Selector for Internet Protocol Quality of Service (IPQoS)

This feature was introduced in the Software Express pilot program and in the Solaris 9 8/03 release.

The Solaris IPQoS feature now includes the user selector, which supplements the existing uid selector. The user selector enables you to specify a user name or userID as criteria in a filter clause in the ipqosconf file. Previously, the uid selector only accepted a userID as a value. The following filter clause from an ipqosconf file shows the user selector:

```
filter {
        name myhost;
        user root;
}
```

For information about filters and selectors, refer to the ipqosconf(1M) man page.

See also the *System Administration Guide: IP Services*.

Routing Information Protocol version 2 (RIPv2)

This feature was introduced in the Software Express pilot program and in the Solaris 9 9/02 release.

Solaris system software now supports Routing Information Protocol version 2 (RIPv2).

RIPv2 adds Classless Inter-Domain Routing (CIDR) and Variable-Length Subnet Mask (VLSM) extensions to the RIPv1 protocol. Message Digest 5 (MD5) extensions protect routers against intentional misdirection by malicious users. The new `in.routed` implementation also includes a built-in Internet Control Message Protocol (ICMP) Router Discovery (RFC 1256) mechanism.

RIPv2 supports multicast if the point-to-point links are enabled with multicast. RIPv2 also supports unicast. If you configure a broadcast address by using the `/etc/gateways` file, then RIPv2 supports broadcast.

For information on how to configure RIPv2, see the `in.rdisc`(1M), `in.routed`(1M), and `gateways`(4) man pages.

Developer Tool Enhancements

This section describes all development tools in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002. DTrace dynamic tracing facility is a feature of particular importance.

Developers should also review these new features in the security and system administration sections:

- "Kernel Modular Debugger" on page 25
- "OpenSSL and OpenSSL PKCS#11 Engine" on page 55

Dynamic Tracing Facility

This feature was introduced in the Solaris Express 11/03 release.

The Solaris DTrace is a comprehensive dynamic tracing facility that gives Solaris users, administrators, and developers a new level of observability into the kernel and user processes. For further information, see "DTrace Dynamic Tracing Facility" on page 18.

GCC Version 3.4.3

This feature was introduced in the Solaris 10 release.

GCC is the open source "C" compiler from the GNU Software Foundation. Build tools include `gmake`, `bison`, `binutils`, `gnuM4`, and `flex`.

Perl Version 5.8.4

This feature was introduced in the Solaris Express 8/04 release.

A new, default version of the Practical Extraction and Report Language (Perl) is available in the Solaris 10 release. The new default version of Perl is version 5.8.4. You need to reinstall any modules that you have manually installed after upgrading to the Solaris 10 software. For more information about installing modules, see "Perl 5" in the *System Administration Guide: Network Services*.

For more information about Perl, see the perl(1) man page.

Per-thread Mode Enhancement

This feature was introduced in the Solaris Express 8/04 release.

The truss, pstack, and pflags commands have been updated to enable the user to specify individual threads within a process or core. Developers and system administrators can better understand the behavior of large multithreaded applications, and target specific threads for debugging.

For further information, see the following man pages:

- truss(1)
- pstack(1)
- pflags(1)

USB End-User Device Support Enhancements

These enhancements were introduced in the Solaris Express 5/04 release. New programming interfaces were added in the Solaris Express 8/04 release.

All current Edgeport USB-to-serial devices now work with this Solaris release. USB 1.1 audio and other isochronous devices now work behind USB 2.0 high-speed hubs.

New Programming Interfaces

These interfaces were introduced in the Solaris Express 8/04 release.

Universal Serial Bus Architecture (USBA) 2.0 programming interfaces are being released publicly as part of the Solaris 10 release. These interfaces are documented as usb_* functions and structures in manual page sections 9F and 9S. See also *Writing Device Drivers*.

The USBA framework now is called USBA 2.0. USB drivers that were written to USBA 1.0 interfaces in prior DDKs are only binary compatible with the Solaris 10 software. These drivers are not source compatible with the Solaris 10 software.

ls Enhancements

This feature is new in the Solaris Express 8/04 release.

The ls command can now display file times to the second or to the nanosecond. See "Enhancements to ls Command" on page 26.

New Functions for Converting Strings

This feature was introduced in the Solaris Express 7/04 release.

Two new functions for converting strings to long integers are now available to kernel module developers. The `ddi_strtol()` and `ddi_strtoul()` functions enable conversion from strings to long integers and unsigned long integers respectively. The new functions provide flexible input, DDI compliancy, and increased error detection.

For more information, refer to the `ddi_strtol`(9F) and `ddi_strtoul`(9F) man pages.

Java Support for `pstack` Command

This feature was introduced in the Solaris Express 7/04 release.

The `pstack` command has been enhanced to display Java frames when using the latest version of Java. For each Java frame in the call stack, the function and line number displays, if available.

See the `pstack`(1) man page.

New Mechanisms for the Solaris Cryptographic Framework

This feature was introduced in the Solaris Express 6/04 release.

The Solaris Cryptographic Framework now supports the following mechanisms for the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols:

- CKM_SSL3_PRE_MASTER_KEY_GEN
- CKM_SSL3_MASTER_KEY_DERIVE
- CKM_SSL3_KEY_AND_MAC_DERIVE
- CKM_SSL3_MASTER_KEY_DERIVE_DH
- CKM_TLS_PRE_MASTER_KEY_GEN
- CKM_TLS_MASTER_KEY_DERIVE
- CKM_TLS_KEY_AND_MAC_DERIVE
- CKM_TLS_MASTER_KEY_DERIVE_DH

The Solaris Cryptographic Framework is an architecture that enables applications in the Solaris system to use or provide cryptographic services. All interactions with the framework are based on the PKCS#11 Cryptographic Token Interface (Cryptoki) from RSA Laboratories, the research arm of RSA Security, Inc.

For more information, see "PKCS #11 Functions: C_GetMechanismList" in the *Solaris Security for Developers Guide*.

Retail and Nonretail Options for Providers in Solaris Cryptographic Framework

This feature was introduced in the Solaris Express 6/04 release.

Vendors of cryptographic providers that connect to the Solaris Cryptographic Framework have more flexibility in requesting certificates from Sun Microsystems. Certificates now support both retail and nonretail export distribution.

Retail encryption products are those products that the U.S. government permits to be shipped to all countries. Retail products cannot be shipped, however, to designated nations that the U.S. government considers to be security threats. Nonretail encryption products are those products that the U.S. government has permitted for domestic use only or for countries that the government has exempted.

For more information, see the elfsign(1) man page and Appendix F, "Packaging and Signing Cryptographic Providers," in the *Solaris Security for Developers Guide*.

Linkers and Libraries Updates

This description was introduced in the Software Express pilot program and revised in the Solaris Express 5/04 release. Linker and libraries updates were also introduced in the Solaris 9 12/02 and subsequent releases.

The Solaris 10 release includes linker-editor features such as string table compression, unreferenced section elimination, and unreferenced dependency detection. For complete information on the most recent feature enhancements, see Appendix D, "Linker and Libraries Updates and New Features," in the *Linker and Libraries Guide*.

Enhancements introduced in the Solaris Express 5/04 release include the following:

- A restructuring of the file system has moved many components from under /usr/lib to /lib. The default search paths for both the link-editor and runtime linkers have been changed accordingly.

- System archive libraries are no longer provided. Therefore, the creation of a statically-linked executable is no longer possible.

- Greater flexibility for defining alternative dependencies is provided with the -A option for the crle command.

Enhancements introduced in the previous Software Express pilot program include the following:

- Greater flexibility in defining the hardware and software requirements of ELF objects is provided with the link-editors.

- The runtime link-auditing interface la_objfilter() has been added.

- Shared-object filtering has been extended to provide filtering on a per-symbol basis.

- Thread-local storage is provided.

- The -z ignore option has been extended to eliminate unreferenced sections during a link-edit. See the ld(1) man page.

- Greater flexibility in defining a symbol's visibility is provided with the "protected" mapfile directive.

- dlopen(3DL) and dlsym(3DL) look-up semantics have been expanded with a new mode, RTLD_FIRST.
- Unreferenced dependencies can be determined by using the ldd utility. See the -U option in the ldd(1) man page.

Layered Driver Interfaces

This feature was introduced in the Solaris Express 3/04 release.

The Solaris 10 Operating System has been enhanced to enable kernel modules to perform device access operations such as open, read, and write a device. This system also enables you to determine which devices are provided through a new set of publicly available Layered Driver Interfaces, referred to as the "LDI."

Driver developers can use the LDI 9F interfaces to access character, block, or STREAMS devices directly from within the Solaris kernel. Application developers can use the LDI 3DEVINFO interfaces to display device layering information. This new architecture also provides administrators with observability into device usage inside the kernel.

The prtconf and fuser utilities have been enhanced to include the following capabilities:

- "Device Layering" through the prtconf command – This command displays device minor node and device usage information. The utility also shows which minor nodes a kernel module currently has open.

    See the prtconf(1M) man page.
- "Device Usage" through the fuser command – This command displays information about users of a device. The command also shows what generic Solaris kernel subsystem or user process opens and accesses a device inside the Solaris kernel.

    See the fuser(1M) man page.

The Layered Driver Interfaces begin with the prefix ldi_. These interfaces are used for device access and to obtain device information at the kernel level. Man pages are provided for the interfaces in section 9F. At the user level, a set of device information library interfaces provides for retrieving kernel device usage information within applications. Man pages are provided for the LDI libdevinfo interfaces within section 3DEVINFO. In addition, the prtconf(1M) and fuser(1M) man pages include information about displaying the kernel device usage information that is provided by the LDI architecture.

For further information, see Chapter 13, "Layered Driver Interface (LDI)," in the *Writing Device Drivers*.

Changes to `makecontext()` Function

This feature was introduced in the Solaris Express 3/04 release and in the Solaris 9 9/04 release.

The semantics of the `uc_stack` member of the `ucontext_t` structure have changes as they apply to inputs to the `makecontext`(3C) `libc` library function. Binary compatibility is preserved between previous versions of Solaris and Solaris 10.

Applications that use this interface must be updated before they are recompiled for the Solaris 10 release. See the `makecontext`(3C) man page for further information.

Single UNIX Specification, Version 3

This feature was introduced in the Solaris Express 2/04 release.

This Solaris release conforms to the Single UNIX Specification, Version 3 (SUSv3). SUSv3 provides updates to POSIX.1-1990, POSIX.1b-1993, POSIX.1c-1996, POSIX.2-1992, and POSIX.2a-1992.

See "Single UNIX Specification, Version 3 Introduces Changes" in the *Solaris 10 Release Notes* for a detailed description of the impact of the SUSv3 updates for Solaris users.

Advanced API

This feature was introduced in the Solaris Express 1/04 release.

The IPv6 Advanced Sockets API updates the Solaris Sockets API to meet the current version of RFC 2292. See "IPv6 Advanced Sockets API" on page 85.

Simple Authentication and Security Layer for Developers

This feature was introduced in the Solaris Express 12/03 release.

Simple Authentication and Security Layer (SASL) provides developers of applications and shared libraries with interfaces for adding authentication, data integrity checking, and encryption to connection-based protocols.

SASL consists of the following items:

- Library, `libsasl`, which provides an API for applications that need authentication, privacy, and integrity services
- Service provider interface (SPI) for third-party plug-ins to add new authentication methods, name canonicalization rules, and property stores
- Header files for development
- Plug-ins that are supplied by Sun for these mechanisms:
  - EXTERNAL
  - PLAIN
  - CRAM-MD5

- DIGEST-MD5
- GSS-API
- GSS-SPNEGO

SASL enables the developer to write to a generic API without having to be concerned about the details of security mechanisms. When developed to use SASL appropriately, servers and clients can use new security mechanisms, naming and user canonicalization plug-ins, and auxprop plug-ins without recompilation.

SASL is described in RFC 2222. SASL is particularly appropriate for applications that use the following protocols that support SASL:

- IMAP
- SMTP
- ACAP
- LDAP

For more information about SASL, see the `libsasl`(3LIB) man page. See also the *Solaris Security for Developers Guide*.

Event Ports

This feature was introduced in the Solaris Express 12/03 release.

Event Ports is a framework that allows applications to generate and collect events from disjoint sources. The framework can retrieve events from multiple objects simultaneously without degrading overall performance.

For further information, see the `port_create`(3C) and `signal.h`(3HEAD) man pages.

Core File Content

The Solaris Express 12/03 release introduced enhancements to the `coreadm`, `gcore`, and `mdb` utilities. See "Core File Content Enhancements" on page 28.

Atomic Operations

This feature was introduced in the Solaris Express 10/03 release.

Atomic operations provide APIs in `libc` that perform simple atomic operations quickly. This new feature permits applications to atomically update memory without using other synchronization primitives or platform-specific assembly language. The available operations include addition, the "and" Boolean function, and the "or" Boolean function.

For more information, see the `atomic_ops`(3C) man page.

Solaris WBEM File Changes

This feature description was updated in the Solaris Express 9/03 release.

There have been several changes to the Managed Object Format (MOF) files in the `/usr/sadm/mof` directory.

- The `Solaris_VM1.0.mof` file was revised to `Solaris_VM2.0.mof` and subsequently to `Solaris_VM3.0.mof`.

- The local file system classes were relocated from `Solaris_VM2.0.mof` to the new file, `Solaris_FS1.0.mof`. `Solaris_FS1.0.mof` defines classes that pertain to storage devices.

- Two of the providers, `Solaris_DiskDrive` and `Solaris_DiskPartition`, that were in the `Solaris_VM1.0.mof` file, were moved into the new `Solaris_DMGT.1.0.mof` file. The `Solaris_DMGT.1.0.mof` file contains classes that represent disks, disk partitions, and other device management classes.

- This release includes another new MOF file, `Solaris_NFS1.0.mof`. The `Solaris_NFS1.0.mof` file defines classes that pertain to NFS devices. This file contains the NFS classes from `Solaris_VM2.0.mof` as well as new classes for configuring and monitoring NFS shares (or "exports") and mounts.

Privileges for Software Developers

This feature was introduced in the Software Express pilot program.

Processes need no longer run as `root` to have superuser capabilities. Instead, superuser capabilities can be parceled out by system administrators as discrete process rights. These process rights are implemented through privileges. Privileges enable developers to limit access to restricted operations and limit the periods for which privileges are in effect. The use of privileges can reduce the harm that formerly resulted if a privileged program was compromised. For compatibility, unmodified programs that run as `root` continue to have all privileges.

For general information about privileges, see "Process Rights Management" on page 52. For information about setting and getting privileges, see the `setppriv`(2) and `getppriv`(2) man pages. To learn more about manipulating privileges, see the `priv_str_to_set`(3C) and the `priv_addset`(3C) man pages.

For further information, see the *Solaris Security for Developers Guide*.

Solaris Cryptographic Framework for Developers

This feature was introduced in the Software Express pilot program.

The Solaris Cryptographic Framework provides cryptographic services to applications. Applications can access the framework through `libpkcs11`(3LIB) and at higher levels.

The Solaris Cryptographic Framework provides the following features for developers of applications that use encryption:

- User-level programming interfaces for various cryptographic functions. These interfaces cover, for example, encryption, decryption, message digests, and signing. The industry standard, RSA Security Inc. PKCS #11 Cryptographic Token Interface (Cryptoki), serves as the API.

   The framework supports the following encryption algorithms:

   - AES
   - DES/3DES
   - RC4
   - MD5
   - SHA-1
   - DSA
   - RSA
   - D-H

- User-level pluggable interfaces for Sun and third-party developers. These interfaces enable administrators to add new plug-ins from providers of encryption algorithms at the user level. Administrators can replace an existing provider with a different implementation. The user service provider interface (SPI) also uses the PKCS#11 standard. Tools for signing, packaging, and installing third-party binaries are provided.

- An optimized software implementation of the most commonly used encryption and digital signing algorithms, such as AES, DES/3DES, and RSA. These implementations have been optimized for the SPARC platform and UltraSPARC platform.

- An administrative CLI tool, `cryptoadm`, for adding or removing encryption plug-ins, setting cryptographic security policy, and other related administrative functions. See the `cryptoadm`(1M) man page.

See the following man pages: `libpkcs11`(3LIB), `pkcs11_softtoken`(5) and `pkcs11_kernel`(5). See also "Solaris Cryptographic Framework for System Administrators" on page 61.

Vendors of software or hardware cryptographic accelerators who are interested in supplying plug-ins to the Solaris cryptographic framework should contact Sun Microsystems for more details.

SPARC: 64-bit Package Changes for Software Developers

This feature was introduced in the Software Express pilot program.

In the Solaris 10 Operating System, packaging has been simplified with most 32-bit components and 64-bit components being delivered in a single package. See "SPARC: 64-bit Package Changes" on page 45.

SPNEGO Pseudo-Mechanism for GSS-API Applications

This feature was introduced in the Software Express pilot program.

This Solaris 10 release includes a new GSS-API "pseudo-mechanism" for negotiating GSS-API security that is based on the SPNEGO protocol (IETF RFC 2478). Simple and Protected GSS-API Negotiation (SPNEGO) is most useful for applications that are based on GSS-API implementations which support multiple security mechanisms. SPNEGO can be applied when two applications use GSS-API to exchange data and do not know which mechanisms are supported by the other application.

SPNEGO is a pseudo-security mechanism that is represented by the following object identifier:

```
iso.org.dod.internet.security.mechanism.snego (1.3.6.1.5.5.2)
```

SPNEGO enables GSS-API peers to determine in-band whether their credentials share common GSS-API security mechanisms. If the mechanisms are shared, then the peers can select a common mechanism to establish the security context.

For further information, see the mech(4) and the mech_spnego(5) man pages. See also the *Solaris Security for Developers Guide*.

Locality Groups

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/03 release.

The *Programming Interfaces Guide* now includes a chapter that explains the interfaces that interact with locality groups (lgroups). These interfaces can be used to help an application efficiently allocate CPU and memory resources. This capability results in improved performance on some systems.

Thread Stacks in `pmap`

This feature was introduced in the Software Express pilot program.

The pmap utility now labels threads' stacks so the stacks can be identified more easily.

See the pmap(1) man page for further information.

New `DOOR_REFUSE_DESC` Flag

This feature was introduced in the Software Express pilot program.

A new flag, DOOR_REFUSE_DESC, was added to the door_create() function. This new flag simplifies writing door servers which do not accept argument descriptors.

For further information, see the door_create(3DOOR) man page.

Stack Check APIs

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

Stack Check APIs allow for advanced interaction with stack-checking compiler support available in the Sun™ ONE Studio. These APIs should be used in applications that are compiled with stack checking enabled, and either manage their own stacks or attempt to detect their own stack overflows.

Developers who maintain their own thread library need to use the `setustack` interface to enable consumers of their library to compile with stack checking enabled.

See the `stack_getbounds`(3C), `stack_setbounds`(3C), and the `stack_inbounds`(3C) man pages.

Enhanced `crypt()` Function for Software Developers

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

The Software Express releases include new extensions to the `crypt()` function and introduce the `crypt_gensalt()` function. These enhancements allow administrators to change the algorithm that is used to obscure users' UNIX login passwords.

Modules are included for MD5 and Blowfish. The MD5 modules are at `crypt_sunmd5` and `crypt_bsdmd5`. The Blowfish module is at `crypt_bsdbf`.

Developers can create new modules for alternate password-obscuring algorithms. Application developers must use the `crypt_gensalt()` function instead of manually generating the salt string for passing to the `crypt()` function.

Modules for alternate algorithms are specified in the `crypt.conf`(4) file. The `module_path` field specifies the path to the shared library object that implements the two required functions:

- `crypt_gensalt_impl()` – Generates the salt string
- `crypt_genhash_impl()` – Generates the encrypted password

For further information, see the `crypt`(3C) and the `policy.conf`(4) man pages.

New Flags for `madvise()` Function

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

The `madvise()` function enables the kernel to optimize access to a user-defined region of memory. This Solaris release includes three new flags for the `madvise()` function:

- `MADV_ACCESS_LWP` – Gives a specified lightweight process (LWP) resource allocation priority
- `MADV_ACCESS_MANY` – Specifies an address range that is intensively used by processes across the machine
- `MADV_ACCESS_DEFAULT` – Resets an address range's access pattern to the system default

For further information on the `madvise()` function, see the `madvise`(3C) man page.

Memory Allocation With `libumem`

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

`libumem` is a user-mode (nonkernel mode) memory allocator library. `libumem` has features that enable you to debug memory leaks and other aberrations that involve memory usage.

This feature is used in the same way that a standard application binary interface (ABI) allocator, such as `malloc()`, is used. A user-mode application requests an arbitrary number of bytes of memory. Then a pointer is returned that is loaded with the address of the allocated memory.

For further information, see the `libumem`(3LIB) man page.

Smart Card Terminal Interfaces

This feature was introduced in the Software Express pilot program and in the Solaris 9 8/03 release.

Solaris smart card interfaces are a set of public interfaces for Smart Card Terminals. Card-terminal vendors can implement these interfaces in a user-level shared library to provide device-level support for their smart card terminals in Solaris. The set of Solaris smart card terminal interfaces is based on card terminal interfaces that are available as part of the Linux Smartcard framework. Card terminal support libraries from Linux can be ported to the Solaris environment with minimum effort. For further information on smart cards, see the *Solaris Smartcard Administration Guide*.

Smartcard Middleware APIs

This feature was introduced in the Software Express pilot program and in the Solaris 9 9/02 release.

The Solaris Smartcard framework now provides low-level middleware APIs. These APIs can be used to exchange data with a smart card by using a smart card reader. The APIs can be used in platforms such as the Sun Blade™ and Sun Ray™ systems. Applications that are written in the Java language, or in C, can use these interfaces.

For more information, see the `libsmartcard`(3LIB) man page and the JavaDocs in `/usr/share/javadoc/smartcard`. See also the *Solaris Smartcard Administration Guide*.

File System Enhancements

This section describes all file system in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002.

NFS version 4

This feature was introduced in the Software Express pilot program. In the Solaris Express 8/04 release, NFS version 4 became the default.

The Solaris 10 release includes the Sun implementation of the NFS version 4 distributed file access protocol. This version is the next logical step in the evolution of NFS. The NFS version 4 protocol, specified in RFC 3530, was created under the auspices of the Internet Engineering Task Force (IETF). This version is designed to be both vendor neutral and operating system neutral.

NFS version 4 integrates file access, file locking, and mount protocols into a single, unified protocol to ease traversal through a firewall and improve security. The Solaris implementation of NFS version 4 is fully integrated with Kerberos V5, also known as SEAM, thus providing authentication, integrity, and privacy. NFS version 4 also enables the negotiation of security flavors to be used between the client and the server. With NFS version 4, a server can offer different security flavors for different file systems.

The Solaris implementation of NFS version 4 includes delegation, a technique by which the server can delegate the management of a file to a client. This technique can reduce the number of round-trip operations because the client is guaranteed that no modification can occur without the server informing the client. The protocol also includes operation compounding, which allows multiple operations to be combined into a single "over-the-wire" request.

For more information about NFS version 4, refer to Chapter 6, "Accessing Network File Systems (Reference)," in the *System Administration Guide: Network Services*.

UFS Logging Enabled by Default

This feature was introduced in the Solaris Express 4/04 release and in the Solaris 9 9/04 release.

Logging is now enabled by default for all UFS file systems except under the following conditions:

- When logging is explicitly disabled
- If insufficient file system space exists for the log

In previous Solaris releases, you had to enable UFS logging manually.

UFS logging packages into one transaction the multiple metadata changes that compose a complete UFS operation. Sets of transactions are recorded in an on-disk log, and then applied to the actual UFS file system's metadata.

UFS logging provides two advantages:

- If the file system is already consistent because of the transaction log, you might not have to run the `fsck` command after a system crash or an unclean shutdown.
- Starting in the Solaris 9 12/02 release, the performance of UFS logging improves or exceeds the level of performance of nonlogging file systems. This improvement can occur because a file system with logging enabled converts multiple updates to the same data into single updates. This capability reduces the number of overhead disk operations that are required.

For more information, see "What's New in File Systems in the Solaris 10 Release?" in the *System Administration Guide: Devices and File Systems*. See also the `mount_ufs`(1M) man page.

Enhancements to the NFS Client

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/03 release.

The following enhancements have improved the performance of the NFS client:

- Restrictions on wire transfer sizes have been relaxed. Now, the transfer size is based on the capabilities of the underlying transport. For example, the NFS transfer limit for UDP is still 32 Kbytes. However, because TCP is a streaming protocol without the datagram limits of UDP, maximum transfer sizes over TCP have been increased to 1 Mbyte.
- Previously, all write requests were serialized by both the NFS client and the NFS server. The NFS client has been modified to permit an application to issue concurrent writes, as well as concurrent reads and writes, to a single file. You can enable this functionality on the client by using the `forcedirectio` mount option. When you use this option, you are enabling this functionality for all files within the mounted file system. You could also enable this functionality on a single file on the client by using the `directio()` interface. Note that unless this new functionality has been enabled, writes to files are serialized. Also, if concurrent writes or concurrent reads and writes are occurring, then POSIX semantics are no longer being supported for that file.

- The NFS client no longer uses an excessive number of UDP ports. Previously, NFS transfers over UDP used a separate UDP port for each outstanding request. Now, by default, the NFS client uses only one UDP reserved port. However, this support is configurable. If the use of more simultaneous ports would increase system performance through increased scalability, then the system can be configured to use more ports. This capability also mirrors the NFS over TCP support, which has had this kind of configurability since its inception.

For further information, see the *System Administration Guide: Network Services*.

Multiterabyte UFS File Systems

Multiterabyte UFS file system support is available only for systems that run a 64-bit kernel. This feature was introduced in the Software Express pilot program and in the Solaris 9 8/03 release.

The Solaris 10 release provides support for multiterabyte UFS file systems on systems that run a 64-bit Solaris kernel. Previously, UFS file systems were limited to approximately 1 terabyte (Tbyte) on both 64-bit systems and 32-bit systems. All UFS file system commands and utilities have been updated to support multiterabyte UFS file systems.

You can initially create a UFS file system that is less than one Tbyte. You can specify that the file system can eventually be grown to a multiterabyte file system by using the newfs -T command. This command sets the inode and fragment density to scale appropriately for a multiterabyte file system.

Support for a multiterabyte UFS file system assumes the availability of multiterabyte LUNs. These LUNS are provided as Solaris Volume Manager volumes, or as physical disks that are greater than one Tbyte.

Features of multiterabyte UFS file systems include the following:

- You can create a UFS file system to a maximum of 16 Tbytes in size.
- You can create a file system that is less than 16 Tbytes, which can later be increased in size to a maximum of 16 Tbytes.
- Multiterabyte file systems can be created on physical disks and on Solaris Volume Manager's logical volumes.
- UFS logging is enabled by default on file systems greater than 1 Tbyte. Multiterabyte file systems benefit from the performance improvements of having UFS logging enabled. Multiterabyte file systems also benefit from the availability of logging because the fsck command might not have to be run when logging is enabled.

Limitations of multiterabyte UFS file systems include the following:

- You cannot mount a file system that is greater than 1 Tbyte on a system that runs a 32-bit Solaris kernel.

- You cannot boot from a file system that is greater than 1 Tbyte on a system that runs a 64-bit Solaris kernel. This limitation means that you cannot put a root (/) file system on a multiterabyte file system.

- These systems do not support individual files greater than 1 Tbyte.

- The maximum number of files per Tbyte of UFS file system is 1 million. This limit is intended to reduce the time it takes to check the file system with the fsck command.

- The maximum quota that you can set on a multiterabyte UFS file system is 2 Tbytes of 1024–byte blocks.

- Using the fssnap command to create a snapshot of a multiterabyte UFS file system is not currently supported.

For more information, see "What's New in File Systems in the Solaris 10 Release?" in the *System Administration Guide: Devices and File Systems*.

Device File System (`devfs`)

This feature was introduced in the Software Express pilot program.

The devfs file system manages devices in the Software Express releases. Users continue to access all devices through entries in the /dev directory. These entries are symbolic links to entries in the /devices directory. The content of the /devices directory is now controlled by the devfs file system. The entries in the /devices directory dynamically represent the current state of accessible devices on the system. These entries require no administration.

The devfs file system provides the following enhancements:

- Operations in the /devices directory result in attaching device entries. Unused device entries are detached.

- System boot performance is increased because only device entries that are needed to boot the system are attached. New device entries are added as the devices are accessed.

For more information, see the devfs(7FS) man page.

Multiterabyte Disk Support With EFI Disk Labels

This multiterabyte disk support is available only for systems that run a 64-bit kernel. This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

The Solaris 10 release provides support for disks that are larger than 1 terabyte (Tbyte) on systems that run a 64-bit Solaris kernel.

The Extensible Firmware Interface (EFI) label provides support for physical disks and virtual disk volumes. The UFS file system is compatible with the EFI disk label, and you can create a UFS file system that is greater than 1 Tbyte. This release also includes updated disk utilities for managing disks that are greater than 1 Tbyte.

However, the SCSI driver, `ssd`, currently supports disks only up to 2 Tbytes. If you need greater disk capacity than 2 Tbytes, use a disk and storage management product such as Solaris Volume Manager to create a larger device.

For more information on using the EFI disk label, see the *System Administration Guide: Devices and File Systems*. This guide contains important information and restrictions. This information concerns using the EFI disk label with existing software products.

The Solaris Volume Manager software can also be used to manage disks that are greater than 1 Tbyte in this Solaris release. See "Multiterabyte Volume Support in Solaris Volume Manager" on page 21.

New Configuration File for the Autofs Environment

This feature was introduced in the Software Express pilot program.

The new configuration file for your autofs environment, `/etc/default/autofs`, provides an additional way to configure your autofs commands and autofs daemons. Now, the same specifications that you would make on the command line can be made in this new configuration file. However, unlike the specifications you would make on the command line, this file preserves your specifications, even during upgrades to your operating system. Additionally, you no longer are required to update critical startup files to ensure that the existing behavior of your autofs environment is preserved.

You can make your specifications by using the following keywords:

- `AUTOMOUNTD_ENV` permits you to assign different values to different environments. This keyword is the equivalent of the `-D` argument for `automountd`.

- `AUTOMOUNTD_NOBROWSE` turns browsing on, or turns browsing off, for all autofs mount points. This command is the equivalent of the `-n` argument for `automountd`.

- `AUTOMOUNTD_TRACE` expands each remote procedure call (RPC) and displays the expanded RPC on standard output. This keyword is the equivalent of the `-T` argument for `automountd`.

- `AUTOMOUNTD_VERBOSE` logs status messages to the console and is the equivalent of the `-v` argument for the `automountd` daemon.

- `AUTOMOUNT_TIMEOUT` sets the duration for a file system to remain idle before the file system is unmounted. This keyword is the equivalent of the `-t` argument for the `automount` command.

- `AUTOMOUNT_VERBOSE` provides notification of `autofs` mounts, unmounts, and other nonessential events. This keyword is the equivalent of the `-v` argument for `automount`.

For more information, refer to the `automount`(1M) and the `automountd`(1M) man pages.

For further information, see the *System Administration Guide: Network Services*.

X11 Windowing Features

This section describes all X11 windowing features in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002.

Xorg X Server

This feature was introduced in the Solaris Express 10/04 release.

The Xorg X server from the X.Org Foundation X11R6.8 open source release is now provided for x86 systems. This server is based on the X.Org and XFree86 open source releases, which are commonly used on Linux and other platforms. This server provides higher performance and support for a wider range of hardware, including third-party drivers written for the XFree86 and Xorg servers. The Xsun server is still provided in this release for users who need features that are not provided in the Xorg server.

For more information, see the Xorg man pages. The following command locates these man pages: **man -M /usr/X11/man Xorg**.

Xfixes Xserver Extension

This feature was introduced in the Solaris Express 8/04 release.

The Xfixes Xserver Extension from the open source community has been added to support comprehensive cursor support for Sec. 508 accessibility.

Xscreensaver Program

This feature was introduced in the Software Express pilot program and in the Solaris 9 8/03 release.

The Xscreensaver program locks your monitor screen after a period of inactivity in order to prevent others from gaining access to your system session. Xscreensaver supports a variety of animation displays that can be used while the screen is locked. This program is used in the Java Desktop System.

See the `xscreensaver`(1) man page for further information.

X Event Interception Extension (XEvIE)

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

XEvIE is a low-level interface that intercepts all the keyboard and mouse events so that these events can be read, consumed, or modified as needed. This X extension enables better integration of assistive technologies, including technologies in the Java Desktop System.

FreeType 2.1.x

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

FreeType 2.1.x is an open source library that provides a simple application programming interface (API). This API can access font content in a uniform way, independently of the file format. Additionally, some format-specific APIs can be used to access special data in the font file.

Further information on FreeType is available at `http://freetype.org`.

Xserver Virtual Screen Enhancement

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

The Xserver Virtual Screen enhancement provides accessibility support for the Java Desktop System. This capability enables support of magnification software on systems that have just one frame buffer.

Xrender Extension

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

The new Xrender feature increases performance for applications, such as the StarOffice software suite, that run on the Solaris Operating System. The Xrender feature provides a modern appearance for these applications. Xrender uses hardware processing for alpha-blending and transparency effects.

Language Support Enhancements

This section describes all language support features in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002.

Unicode Version 4.0 Support

Unicode Version 3.2 support was introduced in the Software Express pilot program. In the Solaris Express 8/04 release, Solaris Unicode locales support the Unicode Version 4.0.

The Unicode Version 4.0 introduces 1226 new characters over the Unicode Version 3.2. This version also includes both normative changes and informative changes as described in "The Unicode Standard 4.0" (ISBN 0-321-18578-1). The standard can be referenced online at: `http://www.unicode.org/standard/standard.html`.

The UTF-8 character representation has been also changed to a more secure form since the UTF-8 Corrigendum that was originally published in the Unicode Version 3.1 and later updated at the Unicode Version 3.2.

This feature also implements the more secure UTF-8 character representation and byte sequences in `iconv` code conversions and OS-level multibyte functions such as the following:

- `mbtowc(3C)`
- `mbstowcs(3C)`
- `mbrtowc(3C)`
- `mblen(3C)`
- `mbsrtowcs(3C)`
- `fgetwc(3C)`
- `mblen(3C)`

For further information, see the *International Language Environments Guide*.

Code Conversions for Internationalized Domain Name Support

This feature was introduced in the Solaris Express 10/03 release and revised in the Solaris Express 8/04 release. For Solaris 9 users, this feature was introduced in the Solaris 9 4/04 release.

Internationalized Domain Name (IDN) enables the use of non-English native language names as host and domain names. To use such non-English host and domain names, application developers must convert such names into ASCII Compatible Encoding (ACE) names in their applications as specified in the RFC 3490. System administrators and end users are also required to use ACE names in existing system files and applications where the networking or system administration applications do not yet support the non-English IDNs.

This feature aids in the conversion by providing the conversion API with various supported option arguments, a dedicated IDN encoding conversion utility, and `iconv` code conversions. Refer to the following man pages for more detail:

- `libidnkit(3LIB)`
- `idn_decodename(3EXT)`
- `idn_decodename2(3EXT)`

- `idn_encodename(3EXT)`
- `idnconv(1)`
- `iconv_en_US.UTF–8(5)`

For further information, see the *International Language Environments Guide*.

New `iconv` Code Conversions

This feature was introduced in the Solaris Express 11/04 release.

Various new `iconv` code conversions between single-byte PC and Windows code pages and various Unicode forms have been added. Also, several major Asian code pages and UCS-2LE have been added.

Details are available at Appendix A in the *International Language Environments Guide*. See also the `iconv_en_US.UTF–8(5)` man page.

New Solaris Unicode Locales

This feature was introduced in the Solaris Express 7/04 release. Additional locales were added in the Solaris 9 9/04 release and in the Solaris 10 release.

The following Unicode locales have been added to Solaris:

- nl_BE.UTF-8 (Belgium Flemish)
- nl_NL.UTF-8 (Dutch)
- ar_SA.UTF-8 (Saudi Arabia)
- el_GR.UTF-8 (Greece)
- pt_PT.UTF-8 (Portugal)
- cs_CZ.UTF-8 (Czech Republic UTF-8)
- cs_CZ.UTF-8@euro (Czech Republic UTF-8 + Euro)
- hu_HU.UTF-8 (Hungary UTF-8)

The new locales are available from system login.

Locale Administrator

This feature was introduced in the Solaris Express 11/03 release and in the Solaris 9 9/04 release.

Locale administrator allows the user to query and configure the locales for a Solaris OS through a command-line interface. Using the `localeadm` tool, a user can display information about locale packages that are installed on the system or that reside on a particular device or directory. The user can add and remove locales on the current system on a per-region basis. For example, the user can add all locales in the Eastern European region to the current system. See the `localeadm(1M)` man page.

Prior to this feature's introduction, after a system was installed the user had to add or remove individual packages to change the locales on the machine. This process is prone to error because it is easy to miss or overlook packages. This tool is a supplement to the locale selection logic in the Solaris installer. The installer is still the primary application for the correct installation of Solaris locales.

Standard Type Services Framework

This feature was introduced in the Solaris Express 9/03 release and in the Solaris 9 4/04 release.

The Standard Type Services Framework (STSF) is a pluggable object-based architecture that allows users to access typographically sophisticated text layout and rendering. The pluggable architecture of the framework gives users the ability to use different font rasterization engines and text layout processors to achieve the desired visual representation. The pluggable architecture also manages fonts and enables application-specific fonts to be created. STSF includes both a standalone API and an X server extension to handle rendering on the server side for improved efficiency. STSF is an open source project that is sponsored by Sun Microsystems.

For more information about the project and how to use the API, see `http://stsf.sourceforge.net`.

Auto Encoding Finder

This feature was introduced in the Solaris Express 9/03 release.

The Auto Encoding Finder is a useful utility for global character handling. Through a general-purpose interface, Auto Encoding Finder provides an easy way to detect the encoding of particular file or string, simplifying access to various language character encodings. For example, the utility simplifies the display of web pages that do not specify encoding information. Search engines, knowledge databases, and machine translation tools might also need to detect the encoding of the language data being accessed. The Auto Encoding Finder tool simplifies this process.

See the `auto_ef`(1) or `libauto_ef`(3LIB) man pages for more detail.

SunIM-based Korean Language Engine With Auxiliary Window Support

This feature was introduced in the Software Express pilot program.

Korean users of the Solaris Operating System can benefit from more comprehensive keyboard input method support for the Korean language. The new Korean Language Engine with auxiliary window support offers Korean users four auxiliary windows to control and configure the Korean input method (IM).

■ User-based preferences can be set within one window.

■ A virtual keyboard environment can be set within another window for point-and-click selection of Korean characters.

- Within another window, users can choose the symbols that they need from special characters that are based on code points.

- Users can organize all of the windows within a special palette of control.

This IM supports three separate keyboard layouts: 2 beol sik, 3 beol sik 390, and 3 beol sik final.

Common Transliteration-based Input Method for All Indian languages

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/03 release.

Users who operate within any Unicode (UTF-8) locale of the Solaris Operating System can now easily and intuitively input characters from Indian regional languages. Users who interact with CDE applications, StarOffice, or Mozilla can more easily interact with Indian scripts. After selecting the transliteration-based input method (IM), users can type phonetic equivalents of Indian language scripts in English. These equivalents are then displayed in the script that is selected, and are correctly shaped and rendered with the help of an underlying layout and shaper module. As transliteration is the most commonly used input method for Indian languages, this support can greatly enhance the usability of the eight Indian scripts that are provided in the Solaris software.

Additional Keyboard Support

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

Software support for three additional keyboards has been added to the Solaris software: Russian keyboard Type 6, Estonian keyboard Type 6, French-Canadian keyboard Type 6, and Polish programmer's keyboard Type 5. This software support gives users in Russia, Canada, Estonia, and Poland greater flexibility for keyboard input. The support modifies standard U.S. keyboard layouts to different language needs.

Wubi Input Method

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

The Wubi input method (IM) is widely used in China. The encoding rule for Wubi IM is based on the radical or stroke shape of Chinese characters. Users can rapidly type Chinese characters through a standard keyboard rather than through slower, phonetic-based input methods.

Input Method Support for Indic

This feature was introduced in the Software Express pilot program.

Input support for Indian regional language keyboards has been added to the Solaris Operating System. Indic language users can type Indic language characters by using their preferred keyboard layouts in the Solaris Operating System.

For further information, see the *International Language Environments Guide*.

Seven Additional Indic Scripts for Support in Unicode Locales

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

In addition to the current support for Hindi, the following Indic scripts are now supported in the Solaris software:

- Bengali
- Gurmukhi
- Gujarati
- Tamil
- Malayalam
- Telugu
- Kannada

Speakers of these Indian regional languages have language support in the Solaris Operating System for any of the Unicode locale environments that Solaris supports.

For further information, see the *International Language Environments Guide*.

`HKSCS-2001` Support in Hong Kong Locales

This feature was introduced in the Software Express pilot program.

`HKSCS-2001` is a new version of the Hong Kong Supplementary Character Set (HKSCS). This new version adds 116 characters to the previous `HKSC-1999` character set. `HKSCS-2001` is supported in these Solaris Hong Kong locales: `zh_HK.BIG5HK` and `zh_HK.UTF-8`.

Additional Software

This section describes addition software features in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002.

Sun Remote Services Net Connect 3.1.1

Sun[SM] Remote Services (SRS) Net Connect 3.1.1 was made available with the Solaris Express 10/04 release on the Extra Value CD. SRS Net Connect 3.1.1 enables you to self-monitor systems, create performance and trend reports, and receive automatic notifications. These features help you maximize system availability and reliability and manage potential issues.

Note – For Solaris 9 users, this feature was introduced in the Solaris 9 4/04 release.

For information about installation of SRS Net Connect, see the *Sun^SM Remote Services Net Connect Activation Guide.*

Freeware Enhancements

This section describes all freeware features in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002.

GCC v.3.4.3

This feature was introduced in the Solaris 10 release. See .

Webmin

This feature was introduced in the Solaris Express 11/04 release.

Webmin is a web-based system administration tool that is written in Perl. Webmin is designed to run on all UNIX and UNIX-like systems, including the Solaris OS. The tool contains many standard modules for the administration of UNIX facilities that are included in the Solaris OS. Webmin also contains modules for administration of facilities that are provided by other open source projects and by commercial vendors. Additional third-party modules are available to administer facilities and products that are not supported by the standard modules.

Webmin is open source software. Documentation is available at `http://www.webmin.com`.

Intelligent Platform Management Interface

This feature was introduced in the Solaris Express 11/04 release.

Intelligent Platform Management Interface (IPMI) is an industry standard that supports some aspects of "lights-out" server management. Most x86-based servers have a built-in management controller (bmc) that enables systems to be remotely managed. For example, a system can be powered on and off, and temperatures and voltages can be measured remotely.

IPMI is open source software. Documentation is available on the web at `http://openipmi.sourceforge.net/`.

Apache Version 2

This feature was introduced in the Solaris Express 11/04 release.

Apache version 2 provides several improvements to Apache version 1, most notably IPv6 support. In addition, SSL/TLS has now been incorporated directly into the server, instead of being provided as a set of add-on patches and modules that must be synchronized. Apache version 1 remains the default web server in the Solaris 10 OS.

Apache version 2 is open source software. Documentation is available at `http://www.apache.org`.

BIND 9

BIND 9 was introduced in the Solaris Express 8/04 release. The Solaris 10 release contains BIND version 9.2.4.

BIND is an open source implementation of DNS. BIND is developed by the Internet Systems Consortium (ISC). BIND allows DNS clients and applications to query DNS servers for the IPv4 and IPv6 networks. BIND includes two main components: a stub resolver API, `resolver`(3resolv), and the DNS name server with various DNS tools.

BIND enables DNS clients to connect to IPv6 DNS servers by using IPv6 transport. BIND provides a complete DNS client-server solution for IPv6 networks.

BIND 9.2.4 is a redesign of the DNS name server and tools by the Internet Systems Consortium (ISC). The BIND version 9.2.4 nameserver and tools are available in the Solaris 10 release.

BIND 8.x-to-BIND 9 migration information is available in the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*. Additional information and documentation about BIND 9 is also available on the ISC web site at `http://www.isc.org`. For information about IPv6 support, see the *System Administration Guide: IP Services*.

Samba Enhancements

In the Solaris Express 8/04 release, Samba, which provides file and print services to SMB/CIFS clients, has been upgraded to version 3.0.4. This version fixes several bugs. Samba was previously updated to version 3.0 in the Solaris Express 3/04 release.

Samba provides seamless file and print services to SMB/CIFS clients. Using Samba, you can share a server's disks and printers with many network clients, including the following:

- LAN Manager
- Windows for Workgroups, Windows 95, 98, and ME
- Windows NT, 2000, and XP
- Linux
- OS/2

Two UNIX clients can access file space and printers on any SMB/CIFS server. The first is an ftp-like client. The second client creates a virtual "`/smb`" directory that provides access to smb-shared directories and printers.

Samba 3.0 updates the SMB file and print server to further improve printing speed. In addition, Samba 3.0 provides the following enhancements:

- Adds single sign-on with Active Directory integration
- Supports international character sets
- Integrates with the LDAP
- Replaces NT primary domain controllers (PDCs) and backup domain controllers (BDCs)

For further information, see the Samba man pages. The following command locates these man pages: man -M */usr/sfw/man samba*. See also *Using Samba, 2nd Edition* at `http://www.oreilly.com/`.

Flex 2.5.4a

This feature was introduced in the Solaris Express 8/04 release.

The `flex` open source lexer is an enhanced version of the `lex`(1) utility. `flex` provides the ability to generate very fast (but large) scanners and batch these scanners. `flex` has the capability to generate scanners with different globally visible variable and function names, as opposed to the `lex`(1) convention of names such as `yy_foo` or `yy_bar`.

For more information, see the man page in the `/usr/sfw/man` directory.

SIP Proxy Server

This feature was introduced in the Solaris Express 8/04 release.

The Session Initiation Protocol (SIP) server provides support for the deployment of VoIP/SIP services by providing a SIP proxy, registrar, and redirect server. This server is called SER. SER is RFC 3261–compliant and is available from `http://iptel.org`.

Further information is provided in the `/usr/sfw/doc/ser` directory and in the man pages in the `/usr/sfw/man` directory.

libusb 0.1.8

This feature was introduced in the Solaris Express 8/04 release.

The libusb library contains a set of interfaces for managing USB devices without a kernel driver.

For further information, see the `libusb` man page in the `/usr/man` directory. This man page points to more information in the `/usr/sfw/share/doc/libusb` directory.

Ghostscript 7.05

This feature was introduced in the Software Express pilot program and in the Solaris 9 8/03 release.

The following new freeware package is included in the Solaris Operating System.

Ghostscript 7.05 – Reads PostScript and PDF files. Displays these files on screen or converts the files to a form that is usable by many printers. See the gs(1) man page under/usr/sfw/share/man.

Note – To view license terms, attribution, and copyright statements for Ghostscript, display the license path in /usr/sfw/share/src/<freeware name>.

New Freeware Packages: libxml2 2.4.16 and libxslt 1.0.19

This feature was introduced in the Software Express pilot program and in the Solaris 9 4/03 release.

The following freeware packages are new or revised in the Software Express releases:

- libxml2 2.4.16 – A standard for building tag-based structured documents or data
- libxslt 1.0.19 – An XML language to define transformation for XML

Note – To view license terms, attribution, and copyright statements for these packages, display the license path in /usr/share/src/<freeware name></filename>.

ANT 1.4.1 Freeware Package

This feature was introduced in the Software Express pilot program and in the Solaris 9 12/02 release.

The following new freeware package is included in the Solaris Operating System: ANT 1.4.1 – This is a Jakarta ANT Java and XML-based build package.

Note – To view license terms, attribution, and copyright statements for ANT, the license path is /usr/sfw/share/src/<freeware name>.

Solaris 10 Documentation Changes

This section describes major changes to the documentation in the Solaris 10 OS that are new or have been enhanced since the Solaris 9 OS was originally distributed in May 2002. These changes include the following:

Solaris 10 Documentation DVD

This feature is new in the Solaris 10 release.

The Solaris 10 release includes all documentation on a new Solaris 10 Documentation DVD. This DVD replaces the CD format that was used for documentation in the Solaris 9 release.

For information about accessing Solaris 10 documentation, including instructions for using the new documentation DVD, see the *Solaris 10 Documentation README*.

New Documents

The following documents are new in the Solaris 10 product.

Device Driver Tutorial
This tutorial provides hands-on information about how to develop device drivers for the Solaris Operating System. This book includes descriptions of writing, building, installing, loading, and testing real device drivers. These instructions help you understand how drivers control devices. This book also gives an overview of the driver development environment, tools available to you to develop drivers, and techniques for avoiding some driver development problems.

Introduction to the Solaris Development Environment
The Solaris Operating System provides developers with numerous interfaces, frameworks, and tools to take advantage of Solaris technologies. This book provides an overview of the Solaris OS, abstracts of key manuals for Solaris developers, and links to sources of detailed information.

Solaris Dynamic Tracing Guide
The Solaris Express 1/03 release introduced the new *Solaris Dynamic Tracing Guide*. The guide includes a complete feature reference and examples for new users. See "DTrace Dynamic Tracing Facility" on page 18 for further information about DTrace.

Solaris Security for Developers Guide
The *Solaris Security for Developers Guide* describes the public application programming interfaces (API) and service provider interfaces (SPI) for the security features in the Solaris OS. This manual is intended for C-language developers who want to write the following types of programs:

- Privileged applications that can override system controls
- Applications that use authentication and related security services
- Applications that need to secure network communications
- Applications that use cryptographic services
- Libraries, shared objects, and plug-ins that provide or consume security services

The manual describes the following public Solaris interfaces for security:

Process Privilege Interfaces	Process privileges permit developers to enable the delegation of security overrides in privileged applications.
PAM	Pluggable authentication modules for the initial authentication of a user to a system.
GSS-API	Generic security service application program interface, for secure communication between peer applications. GSS-API provides authentication, integrity, and confidentiality protection services as well.
SASL	Simple authentication and security layer, used largely by protocols for authentication, privacy, and data integrity. SASL is intended for higher-level network-based applications.
Cryptographic Framework	A framework based on standard PKCS #11 interfaces that accommodates consumers and providers of cryptographic services.
Smart Card	A set of interfaces for developers of IFD handlers for smart card terminals.

Working examples are provided.

Solaris System Management Agent Administration Guide
The System Management Agent (SMA) is based on the open source Net-SNMP agent. This manual is for administrators who want to use the System Management Agent to manage network devices securely, and to migrate their SNMP solution from the Solstice Enterprise Agents software to the System Management Agent. This manual includes a chapter that covers security topics with examples provided.

Solaris System Management Agent Developer's Guide
The System Management Agent (SMA) is based on the open source Net-SNMP agent. This manual provides information for developers who want to create MIB modules to extend the functionality of the agent.

System Administration Guide: Solaris Containers—Resource Management and Solaris Zones
The Solaris Express 2/04 release introduced the new *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*. This book is for anyone responsible for administering one or more systems that run the Solaris 10 release. The book covers resource management topics such as projects and tasks, extended accounting, resource controls, and dynamic resource pools. This book also covers

virtualization with Solaris Zones. See "Solaris Zones Software Partitioning Technology" on page 35 and all descriptions in "System Resources Enhancements" on page 35.

x86 Assembly Language Reference Manual
Introduced in the Solaris Express 6/04 release, the *x86 Assembly Language Reference Manual* documents the syntax of the Solaris x86 assembly language. This manual is provided to help experienced assembly language programmers understand disassembled output of Solaris compilers. This manual is neither an introductory book about assembly language programming nor a reference manual for the x86 architecture.

Document Reorganizations

The following documentation has been reorganized in the Solaris 10 product.

- "Resource Management Documentation Changes" on page 121
- "System Administration Guides Reorganized" on page 121
- "Changes to the *Solaris Installation Guide*" on page 122

Resource Management Documentation Changes

The resource management chapters are in the new volume, *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*. The Solaris 9 book, *System Administration Guide: Resource Management and Network Services*, is now *System Administration Guide: Network Services* in the Solaris 10 OS.

System Administration Guides Reorganized

The Solaris 10 system administration guides have been reorganized.

The content in the *System Administration Guide: Basic Administration* has been reduced to include only the following topics:

- Using Solaris Management Console Tools
- Managing Users Accounts and Groups
- Managing Server and Client Support
- Shutting Down and Booting a System
- Managing Software
- Managing Solaris Patches

Complex device and file system topics were moved to the *System Administration Guide: Devices and File Systems*. This new guide includes device and file system administration topics including the following:

- Managing Removable Media
- Managing Devices
- Managing Disks
- Managing File Systems
- Backing Up and Restoring File Systems

The *System Administration Guide: Advanced Administration* contains the same topics from the previous Solaris releases.

Changes to the *Solaris Installation Guide*

The contents of the *Solaris Installation Guide* have been broken into five books that cover specific topics. For information about the content of each new book, see the following descriptions.

- *Solaris 10 Installation Guide: Basic Installations*

    This book describes a basic OS installation with a graphical user interface (GUI).

- *Solaris 10 Installation Guide: Network-Based Installations*

    This book describes how to perform a remote Solaris installation over a local area network or a wide area network.

- *Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning*

    This book provides planning information for using CD or DVD media to upgrade a system to the Solaris Operating System. This book also describes how to use Solaris Live Upgrade to create and upgrade new boot environments.

- *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*

    This book provides instructions for creating Solaris Flash archives and for using Solaris Flash archives to install the Solaris OS on multiple systems.

- *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*

    This book describes how to create the files and directories necessary to perform an unattended custom JumpStart installation. How to create RAID-1 volumes during a JumpStart installation is also described.

Documentation That Has Been Moved

These documents were once published as part of the Solaris 9 release. The documents are now either published elsewhere, or their content has been moved into a different document.

- *About Solaris 9 Documentation* information has been moved to the *Solaris 10 Documentation README* and to this manual.

- *man pages section 3: Realtime Library Functions* information has been moved from *man pages section 3: Threads and Realtime Library Functions* to *man pages section 3: Basic Library Functions*.

- The printed *Solaris 9 Installation Release Notes* is available instead in the online *Solaris 10 Release Notes*.

- The *GSS-API Programming Guide*, which was available in previous Solaris releases, has been subsumed by the new *Solaris Security for Developers Guide*.

Documentation That Is Not Published in This Release

These documents are not published as part of the Solaris 10 product. The documents were previously published as part of the Solaris 9 release and can be found at `http://docs.sun.com`.

- All books in the *CDE Developer Collection*
- All books in the *KCMS Collection*
- *Federated Naming Server Programming Guide*
- *Solstice Enterprise Agents 1.0 User Guide*

Features by Release Date

Many of the features in the Solaris 10 OS were introduced in the Software Express program. This section lists all feature titles by original release date. Each feature title links to its full description.

Note – For a summary of features introduced in the Solaris 9, Solaris 8, or Solaris 7 releases, see *What's New in the Solaris 9 Operating Environment* at `http://docs.sun.com`. This book covers Solaris 9 features, and Solaris 8 and Solaris 7 features are fully described in the appendix.

New Features in Solaris 10 Release

The Solaris 10 release introduced the following new features:

- "Signing ELF Objects" on page 52
- "Virtual IP Source Address Selection" on page 78
- "GCC Version 3.4.3" on page 91
- "Solaris 10 Documentation DVD" on page 119

The Solaris 10 release includes enhancements to the following features:

- "Predictive Self-Healing" on page 16
- "Java Desktop System, Release 3" on page 33
- "BIND 9" on page 116
- "New Solaris Unicode Locales" on page 111

New Features in Solaris Express 11/04 Release

The Solaris Express 11/04 release introduced the following new features:

- "64-bit Support on x86 Systems" on page 19
- "Kernel Selection for x86 Systems" on page 19
- "New `iconv` Code Conversions" on page 111
- "Webmin" on page 115
- "Intelligent Platform Management Interface" on page 115
- "Apache Version 2" on page 115

The Solaris Express 11/04 release included enhancements to the following features:

- "DTrace Dynamic Tracing Facility" on page 18
- "System V IPC and Other Resource Controls" on page 36
- "New or Updated Drivers" on page 69

New Features in Solaris Express 10/04 Release

The Solaris Express 10/04 release introduced the following new features:

- "Solaris Service Manager" on page 16, a new component for "Predictive Self-Healing" on page 16
- "Sun Java Web Console" on page 20
- "Multi-Owner Disk Set Support With Solaris Volume Manager for Sun Cluster" on page 20
- "Importing Remotely Replicated Disk Sets With Solaris Volume Manager" on page 21
- "Device ID Changes" on page 21
- "Package and Patch Tool Enhancements" on page 22
- "Enhancements for `pbind` and `psrset` Commands" on page 23
- "Java Desktop System, Release 3" on page 33
- "`pam_ldap` Changes" on page 54
- "Solaris Secure Shell Enhancements" on page 55
- "`sshd` Daemon and `/etc/default/login`" on page 56
- "New Password Options for Nonlogin and Locked Accounts" on page 56
- "`-setcond` Option to `auditconfig` Command Is Removed" on page 56
- "New or Updated Drivers" on page 69
- "1394 (FireWire) and Mass-Storage Support on x86 Systems" on page 71
- "Internet Printing Protocol Listener" on page 72
- "Sun Remote Services Net Connect 3.1.1" on page 114
- "Xorg X Server" on page 108

The Solaris Express 10/04 release included the following updated features:

- "Predictive Self-Healing" on page 16
- "System V IPC and Other Resource Controls" on page 36

New Features in Solaris Express 8/04 Release

The Solaris Express 8/04 release introduced the following new features:

- "DHCP Event Scripting" on page 24
- "DHCP for Logical Interfaces" on page 24
- "x86: SunVTS 6.0" on page 25
- "Kernel Modular Debugger" on page 25
- "Solaris Process Accounting and Statistics Improvements" on page 25
- "Enhancements to `ls` Command" on page 26
- "Java 2 Platform, Standard Edition 5" on page 40
- "OpenSSL and OpenSSL PKCS#11 Engine" on page 55
- "`perzone` Audit Policy" on page 56
- "Fibre Channel Connectivity for Storage Devices" on page 72

- "Stream Control Transmission Protocol" on page 79
- "Zebra Multiprotocol Routing Suite" on page 79
- "IPsec and NAT Traversal" on page 79
- "Enhancement to the nfsmapid Daemon" on page 80
- "sendmail Version 8.13" on page 81
- "Per-thread Mode Enhancement" on page 92
- "Perl Version 5.8.4" on page 91
- "BIND 9" on page 116
- "Samba Enhancements" on page 116
- "Flex 2.5.4a" on page 117
- "SIP Proxy Server" on page 117
- "libusb 0.1.8" on page 117

The Solaris Express 8/04 release included revisions to the following features:

- "System V IPC and Other Resource Controls" on page 36
- "USB End-User Device Support Enhancements" on page 92
- "NFS version 4" on page 103
- "Unicode Version 4.0 Support" on page 110
- "Code Conversions for Internationalized Domain Name Support" on page 110

New Features in Solaris Express 7/04 Release

The Solaris Express 7/04 release introduced the following new features:

- "New Solaris Project and Resource Management Command Functionality" on page 37
- "New Functions for Converting Strings" on page 93
- "Java Support for pstack Command" on page 93
- "New Solaris Unicode Locales" on page 111

New Features in Solaris Express 6/04 Release

The Solaris Express 6/04 release introduced the following new features:

- "Predictive Self-Healing" on page 16
- "Sun Patch Manager Enhancements" on page 26
- "New psrinfo Option to Identify Chip Multithreading Features" on page 27
- "Enhancements to pfiles Tool" on page 27
- "Solaris Installation Changes Including Installation Unification" on page 43
- "Changes to PAM for the Solaris 10 OS" on page 53
- "Expanded Printer Support" on page 72
- "Using CacheFS With NFS version 4" on page 84
- "New Mechanisms for the Solaris Cryptographic Framework" on page 93
- "Retail and Nonretail Options for Providers in Solaris Cryptographic Framework" on page 93

Also in the Solaris Express 6/04 release, important revisions have been made to the following:

- "Kerberos Enhancements" on page 57
- "Sun Java System Message Queue" on page 82
- "Sun Java System Application Server" on page 83

New Features in Solaris Express 5/04 Release

The Solaris Express 5/04 release introduced the following new features:

- "Common Solaris Target Disk Driver" on page 73
- "Enhancement for vacation Utility" on page 84
- "USB End-User Device Support Enhancements" on page 92

Also in the Solaris Express 5/04 release, revisions have been made to the following:

- "DTrace Dynamic Tracing Facility" on page 18
- "Kerberos Enhancements" on page 57
- "Linkers and Libraries Updates" on page 94

New Features in Solaris Express 4/04 Release

The Solaris Express 4/04 release introduced the following new features:

- "Mozilla 1.7" on page 35
- "TCP Wrappers for rpcbind" on page 58
- "CPU Performance Counters" on page 68
- "Wheel Mouse Support" on page 74
- "MILTER, sendmail's New Mail Filter API" on page 85
- "UFS Logging Enabled by Default" on page 103

New Features in Solaris Express 3/04 Release

The Solaris Express 3/04 release introduced the following new features:

- "Custom JumpStart Installation Package and Patch Enhancements" on page 44
- "Layered Driver Interfaces" on page 95
- "Changes to makecontext() Function" on page 96
- "Samba Enhancements" on page 116
- "Solaris 10 Documentation Changes" on page 119

In addition, enhancements in this release have been made to "System V IPC and Other Resource Controls" on page 36.

New Features in Solaris Express 2/04 Release

The Solaris Express 2/04 release introduced the following new features:

- "Solaris IP Filter" on page 27
- "Solaris Zones Software Partitioning Technology" on page 35
- "zonename Audit Token and Audit Policy Option" on page 59
- "Single UNIX Specification, Version 3" on page 96

New Features in Solaris Express 1/04 Release

The Solaris Express 1/04 release introduced the following new features:

- "Expanded Disk Set Support in the Solaris Volume Manager" on page 21
- "Dynamic Resource Pools" on page 38
- "User Commands for Solaris Cryptographic Framework" on page 59
- "IKE Configuration Parameters" on page 59
- "IPv6 Advanced Sockets API" on page 85
- "Contents of `/usr/lib/mail` Have Moved to `/etc/mail/cf`" on page 85

The following features were enhanced in the Solaris Express 1/04 release:

- "Core File Content Enhancements" on page 28
- "Linkers and Libraries Updates" on page 94

New Features in Solaris Express 12/03 Release

The Solaris Express 12/03 release introduced the following new features:

- "Core File Content Enhancements" on page 28
- "System Management Agent" on page 29
- "LDAP Command Changes" on page 32
- "x86: Specifying Boot Properties With `add_install_client` Command" on page 45
- "Configuring Multiple Network Interfaces During Installation" on page 45
- "Audit Time Now Reported in ISO 8601 Format" on page 60
- "Simple Authentication and Security Layer for Developers" on page 96
- "Event Ports" on page 97

New Features in Solaris Express 11/03 Release

The Solaris Express 11/03 release introduced the following new features:

- "DTrace Dynamic Tracing Facility" on page 18
- "Basic Audit and Reporting Tool" on page 61
- "Added IPv6 Functionality During Solaris Installation" on page 85
- "IPv6 Temporary Addresses" on page 86
- "Locale Administrator" on page 111

New Features in Solaris Express 10/03 Release

The Solaris Express 10/03 release introduced the following new features:

- "New Architecture for Network Stacks" on page 67
- "Atomic Operations" on page 97
- "Code Conversions for Internationalized Domain Name Support" on page 110

New Features in Solaris Express 9/03 Release

The Solaris Express 9/03 release introduced the following new features:

- "Top-Down Volume Creation in Solaris Volume Manager" on page 22
- "Replacement of Solaris Installation Command-Line Interface" on page 45
- "IPsec and the Solaris Cryptographic Framework" on page 61
- "Solaris Support for USB Devices" on page 75
- "EHCI and OHCI Drivers" on page 77
- "sendmail Version 8.12 Uses TCP Wrappers" on page 81
- "routeadm Command" on page 86
- "Standard Type Services Framework" on page 112
- "Auto Encoding Finder" on page 112

Feature descriptions that were revised in the Solaris Express 9/03 release include the following:

- "USB 2.0 Features" on page 74
- "USB 2.0 Devices" on page 74
- "USB Mass Storage Devices" on page 75
- "USB Driver Enhancements" on page 77
- "Sun Java System Application Server" on page 83
- "Solaris WBEM File Changes" on page 98

New Features in Software Express Pilot Program

All features that are described in this book and are not listed in the preceding sections were introduced as part of the Software Express pilot program.