What's New in Oracle® Solaris 11.4

Part No: E60974

Draft -- July 09, 2018

This document summarizes features that are new or have been enhanced in the Oracle Solaris 11.4 release.

Key Features in Oracle Solaris 11.4

Oracle Solaris is a secure, fast platform engineered for large-scale enterprise deployments. Oracle Solaris provides simple update, compliance monitoring, performance monitoring, and zero overhead virtualization for isolating mission-critical workloads.

Proven and automated security and compliance:

- Key Management Interoperability Protocol (KMIP) – Oracle Solaris KMIP cryptography clients can communicate with remote KMIP servers, where the keys are created and managed in secure data-at-rest systems.
- Silicon Secured Memory (SSM) – Automatically protect your application and operating system with SSM.
- Modernize security audits with built-in, automated compliance reporting.
- Secure your application platform deployment with a new application sandbox management tool.

Optimized:

- Fastest and most secure platform for running Oracle Database and Java.
- Maximize Oracle Database multitenant security and isolation.

Simplicity:

- Seamlessly migrate traditional workloads to VMs.
- Use `cloudbase-init` to initially configure guest OS deployments.
- Use the Oracle Solaris Web Dashboard Analytics application to view system data, performance, and compliance.

Bundled software updates. See "Oracle Solaris 11.4 Bundled Software Updates" in *Freeware Available in Oracle Solaris 11.4* for more information:

- New versions of many bundled software packages including GNOME 3.24, ISC BIND 9.10, MySQL 5.7, Open Fabric Enterprise Distribution 3.18, Oracle Instant Client 12.2, Perl 5.26, Puppet 5.5, Python 3.5, and Xorg 1.19.
- Newly bundled software including Augeas, Cython, the `cx_Oracle` Python module, the Google Go compiler, LLVM/Clang, MCollective, Oracle Database Programming Interface-C (ODPI-C), and the `paps` print filter.

Oracle Solaris 11.4 supports the following Oracle application updates:

- Oracle VM Server for SPARC 3.6. See Oracle VM Server for SPARC and Oracle VM Server for SPARC 3.6 Documentation Library.
- Oracle Solaris Cluster 4.4. See Oracle Solaris Cluster and Oracle Solaris Cluster 4.4 Information Library.

Security and Compliance Features

This section describes the security and compliance features that are new in this release. These new features help prevent new threats through anti-malware protection and enable you to meet the strictest compliance obligations.

Secure Sandboxes

Sandboxes are uniquely named sets of process attributes that can be used to specify security and resource isolation requirements. In Oracle Solaris 11.4, you can execute untrusted processes in temporary sandboxes. Persistent and hierarchical sandboxes can be created by using the `sandboxadm` command. Both temporary and persistent sandboxes can be entered by using the `sandbox` command.

Sandboxes are suitable for constraining both privileged and unprivileged applications. Enhanced exploit mitigation controls leveraging SPARC Silicon Secured Secure Memory automatically protect key applications and the system kernel.

For more information, see "Configuring Sandboxes for Project Isolation" in *Securing Users and Processes in Oracle Solaris 11.4* and the `sandboxing`(7), `sandbox`(1), and `sandboxadm`(8) man pages.

Security Compliance Assessment

This Oracle Solaris release supports the ability to run and store compliance reports remotely, and to filter reports according to metadata.

- You can run and administer compliance reports from a central system and store them on a common server. See Chapter 2, "Centrally Managing Compliance Assessments" in *Oracle Solaris 11.4 Compliance Guide* and the `compliance-roster`(8) man page.
- You can tag compliance assessments for identification and filtering. See "Using Metadata to Manage Assessments" in *Oracle Solaris 11.4 Compliance Guide* and the *Match Parameters* section of the `compliance`(8) man page.

Oracle Solaris Cluster Compliance Checks

The standard benchmark for the Oracle Solaris `compliance` command includes checks for Oracle Solaris Cluster. The Oracle Solaris Cluster checks run only when the system has Oracle Solaris Cluster installed and configured.

See the following documentation for information about benchmarks, profiles, the `compliance` command, and Oracle Solaris Cluster compliance checks:

- `compliance`(8) man page
- "What's New in Compliance in Oracle Solaris 11.4" in *Oracle Solaris 11.4 Compliance Guide*
- *Oracle Solaris Cluster 4.4 Security Guidelines*

Per File Auditing

Per file auditing in Oracle Solaris 11.4 provides fine-grained, on-access auditing of specific files and directories. With this feature, system and security administrators can target specified files to be audited. The specified files can be accessed in certain ways, allowing for much easier collection and analysis of audit data.

For example:

```
# chmod A+everyone@:write_data/read_data:successful_access/failed_access:audit /data/db1
```

This audit ACE ensures that an audit record is generated for any reads or writes, both success and denied access, on the `/data/db1` file by any user on the system. Audit ACEs can also be added for metadata changes.

For more information, see "What's New in the Audit Service in Oracle Solaris 11.4" in *Managing Auditing in Oracle Solaris 11.4*.

Verified Boot Auditing

In Oracle Solaris 11.4, this new feature helps you generate audit records to indicate the signature verification results of the kernel modules. The feature checks the Verified Boot `boot_policy` value when Oracle Solaris 11.4 boots, and outputs the value to an audit record for `AUE_SYSTEMBOOT` event. When Verified Boot is enabled with the value of `boot_policy` property as `warning` or `enforce`, Oracle Solaris audit produces `AUE_MODLOAD` audit events if an elfsign signature verification fails when a module is to be loaded. With Verified Boot enabled, you can keep track of events for kernel modules that have invalid signatures or signatures that have not been loaded into the system.

For more information, see "New Feature – Auditing Verified Boot" in *Managing Auditing in Oracle Solaris 11.4*.

Privileged Command Execution History Reporting

Oracle Solaris 11.4 introduces the `admhist` utility, which is used to provide a summary of system administration related events that have been run on the system, in a helpful, easy-to-understand format. The `admhist` utility leverages audit data that enables the `praudit` and `auditreduce` utilities to provide more detailed log analysis.

A variety of options are available that enable you to narrow the results by user, date, time, or type of event as follows. For example, you can identify privileged command executions by a particular user ID within the last 24 hours:

```
# admhist -v -a "last 24 hours"
2017-05-09 10:58:55 user1@example.com cwd=/export/home/user1 /usr/sbin/zfs get quota rpool/export/
home/user1
2017-05-09 10:59:16 user1@example.com cwd=/export/home/user1 /usr/sbin/zfs set quota 40g
2017-05-09 10:59:27 user1@example.com cwd=/export/home/user1 /usr/sbin/zfs get quota rpool/export/
home/user1
2017-05-09 10:59:31 user1@example.com cwd=/export/home/user1 /usr/bin/bash
2017-05-09 10:59:31 user1@example.com cwd=/ /usr/bin/su
```

The output illustrates that the user `user1` switched to the `root` user and increased his quota. The privileges that are used throughout the life of the process are examined when the command exits, which is why the `su` operation is listed at the end of the output.

For more information, see the `admhist`(8) man page, "New Feature – Per-Privilege Logging of Audit Events" in *Managing Auditing in Oracle Solaris 11.4*, and *Using Oracle Solaris 11.4 Analytics*.

KMIP Client Support

Oracle Solaris 11.4 provides client support for using the Key Management Interoperability Protocol (KMIP) version 1.1. A new PKCS#11 provider, `pkcs11_kmip`, is provided in the Oracle Solaris Cryptographic Framework, which enables PKCS#11 applications to function as KMIP clients and communicate to KMIP-compliant servers.

Oracle Solaris 11.4 also includes a new command, `kmipcfg`, which initializes and manages the states of the `pkcs11_kmip` provider.

For more information, see Chapter 5, "KMIP and PKCS #11 Client Applications" in *Managing Encryption and Certificates in Oracle Solaris 11.4* and the `pkcs11_kmip`(7) and `kmipcfg`(8) man pages.

File and Process Labeling

File and process labeling in Oracle Solaris 11.4 provides a framework for restricting access to sensitive information. Files and directories can now be labeled to provide access to users or roles with sufficient `clearance`. The `clearance` policy also applies to processes with all privileges. Oracle Solaris 11.4 can generate logs of every access to labeled files, which can be used to meet compliance standards such as `PCI-DSS` and `HIPAA`.

For more information, see "Labels and Clearances" in *Securing Files and Verifying File Integrity in Oracle Solaris 11.4* and the `clearance`(7) man page.

Silicon Secured Memory Security Exploit Mitigations

Silicon Secured Memory (SSM) adds real-time checking of access to data in memory to help protect against malicious intrusion and flawed program code in production for greater security and reliability.

SSM is available via the default system memory allocator and is available inside a kernel zone. See "Silicon Secured Memory Support in Oracle Solaris Kernel Zones" on page 19.

The system default allocator (`libc malloc`) is now Application Data Integrity (ADI) aware. Binaries tagged with the `sxadm` command automatically receive the protection. See the `ADIHEAP` and `ADISTACK` protections in the Security Extensions section of the `sxadm`(8) man page.

SSM application programming interfaces are available for advanced customization. See "Protecting Against Malware With Security Extensions" in *Securing Systems and Attached Devices in Oracle Solaris 11.4* and the `adi`(2) man page.

Packet Filter

Oracle Solaris 11.4 includes the OpenBSD Packet Filter (PF) firewall for filtering TCP/IP traffic. PF firewall is a replacement to the IP Filter (IPF) in Oracle Solaris 11.4, enabling both bandwidth management and packet prioritization. To use the PF firewall, install the `pkg:/network/firewall` package and enable the `svc:/network/firewall:default` service instance.

Note - Make sure you configure the firewall before enabling the service. The default configuration puts the service to a degraded state. The degraded firewall blocks all inbound sessions except `ssh`. Outbound sessions are allowed.

PF includes the `pflogd` feature, a packet logging daemon that safely saves packets logged by the PF firewall. These packets are available from a capture datalink. The daemon reads packets from this datalink and stores them into a file. For more information, see the `pflogd`(8) man page.

PF supports `ftp-proxy`, a semi-transparent proxy for FTP, supporting IPv4 NAT. Systems running the PF firewall for NAT can use the `ftp-proxy` to allow FTP connections to pass through the firewall. For more information, see the `ftp-proxy`(8) man page.

For more information, see Chapter 3, “Oracle Solaris Firewall” in *Securing the Network in Oracle Solaris 11.4* and the `pfctl`(8), `pf.conf`(7), and `pf.os`(7) man pages.

Kerberos

Oracle Solaris 11.4 provides an updated version of Kerberos, which includes improvements from the latest version of MIT Kerberos, as well as enhancements made for Oracle Solaris. Kerberos provides network authentication, and optionally provides message integrity and privacy, depending on how an application uses it.

For more information, see Chapter 1, “Kerberos on Oracle Solaris” in *Managing Kerberos in Oracle Solaris 11.4* and the `kerberos.7` man page.

`libsasl2`

The Simple Authentication and Security Layer (SASL) framework provides authentication and optional security services for network protocols. Oracle Solaris 11.4 bases its SASL implementation on the open source Cyrus SASL version 2.1.26 with a few changes.

The SASL plugins are in the `/usr/lib/sasl2` directory, and the default location for the SASL configuration files is the `/etc/sasl2` directory. By basing the SASL version on open source, Oracle Solaris 11.4 is able to provide the latest SASL features, including security updates.

For more information, see Chapter 2, “Using Simple Authentication and Security Layer” in *Managing Authentication in Oracle Solaris 11.4*.

`libucrypto` Library

Oracle Solaris 11.4 includes `libucrypto`, a lightweight library that provides access to hardware accelerated cryptography. Operations provided include symmetric and asymmetric encryption, digital signatures, message authentication codes, and cryptographic hashes. The `libucrypto` library provides lightweight access to hardware cryptographic primitives, when you do not need access to key storage, session management, or the standards based APIs provided by `libpkcs11`.

The `libucrypto` library enables fast access to hardware-accelerated cryptography. The library is fast for both the programmer and the processor, as it avoids locking and session management overhead.

For more information, see “Simple and Fast ucrypto Provider” in *Managing Encryption and Certificates in Oracle Solaris 11.4* and the `libpkcs11`(3LIB) man page.

`account-policy` Service

This Oracle Solaris release offers an alternative to editing individual files in the `/etc` directory to establish system policy. The `account-policy` Service Management Facility (SMF) service stores `login`, `su`, shell variables, logging, security policy (`policy.conf`), and RBAC settings as properties in SMF. When the service is enabled, you set and get system policy through the service. Note that the `/etc` files might not indicate the policies that are in effect. For more information, see the `account-policy`(8S) man page and “Modifying Rights System-Wide As SMF Properties” in *Securing Users and Processes in Oracle Solaris 11.4*.

PKCS #11 v2.40 Support for Oracle Solaris Cryptographic Framework

The Oracle Solaris Cryptographic Framework has been updated from PKCS #11 v2.20 to PKCS #11 v2.40. The updates include some of the latest mechanisms in PKCS #11 v2.40 including those from PKCS #11 v2.30. A new error code and a new value have also been introduced in PKCS #11 v2.40. The following new mechanisms have been added:

- AES signing and verification

  ```
  CKM_AES_XCBC_MAC
  CKM_AES_XCBC_MAC_96
  CKM_AES_CMAC
  CKM_AES_GMAC
  ```

- AES encryption and decryption

  ```
  CKM_AES_GCM
  CKM_AES_CCM
  CKM_AES_CFB128
  ```

- SHA-512/t message digesting

  ```
  CKM_SHA512_224
  CKM_SHA512_256
  CKM_SHA512_T
  ```

- SHA-512/t general-length with HMAC

  ```
  CKM_SHA512_224_HMAC_GENERAL
  CKM_SHA512_256_HMAC_GENERAL
  CKM_SHA512_T_HMAC_GENERAL
  CKM_SHA512_224_HMAC
  CKM_SHA512_256_HMAC
  CKM_SHA512_T_HMAC
  ```

- SHA-512/t key derivation

  ```
  CKM_SHA512_224_KEY_DERIVATION
  CKM_SHA512_256_KEY_DERIVATION
  CKM_SHA512_T_KEY_DERIVATION
  ```

- TLS 1.2

  ```
  CKM_TLS12_MASTER_KEY_DERIVE
  CKM_TLS12_MASTER_KEY_DERIVE_DH
  CKM_TLS12_KEY_AND_MAC_DERIVE
  CKM_TLS12_KEY_SAFE_DERIVE
  CKM_TLS_KDF - replacing CKM_TLS_PRF
  CKM_TLS_MAC - replacing CKM_TLS_PRF
  ```

- Error code `CKR_CURVE_NOT_SUPPORTED` for elliptic curve

  If a specific elliptic curve cannot be supported, then the error code `CKR_CURVE_NOT_SUPPORTED` is returned. In the previous version, `CKR_TEMPLATE_INCONSISTENT` was returned if the curve was not supported.

- `CK_UNAVAILABLE_INFORMATION`

  When `C_GetAttributeValue()` is called, and if an attribute cannot be returned because of its invalidity or unavailability, `ulValueLen` is set to `CK_UNAVAILABLE_INFORMATION`. The caller has to check if the returned attribute value is invalid or unavailable by comparing `ulValueLen` with `CK_UNAVAILABLE_INFORMATION`. Moreover, the caller has to treat `ulValueLen = 0` as a valid value.

- Attributes `CKA_DESTROYABLE` and `CKR_ACTION_PROHIBITED`

  If an object has `CKA_DESTROYABLE = CK_FALSE`, then a request to `C_DestroyObject` for this particular object should result in `CKR_ACTION_PROHIBITED` being returned as error code.
- Removing Restrictions with `CKU_SO`

  This change removes the restrictions on having R/O open while `CKU_SO` is logged in. While R/O sessions can now co-exist with `CKU_SO`, those sessions behave as `CKS_RO_PUBLIC_SESSION`. An R/O session cannot be used to `C_Login` with `CKU_SO`.

  `CKR_SESSION_READ_ONLY_EXISTS` and `CKR_SESSION_READ_WRITE_SO_EXISTS` are deprecated.

For more information, see the `SUNW_C_GetMechSession`(3EXT), `SUNW_C_KeyToObject`(3EXT), `libpkcs11`(3LIB), `pkcs11_softtoken`(7), `pkcs11_kms`(7), and `pkcs11_tpm`(7) man pages.

Data Management Features

This section describes the data management features that are new in this release. These features enable you to scale out design with unlimited capacity for future growth and also provide enhanced data integrity.

See also "Configure Immutable Zones by Running in the Trusted Path" on page 18.

ZFS Top-Level Device Removal

The `zpool remove` command enables you to remove top-level data devices. Removing a top-level data device migrates the data from the device to be removed to the remaining data devices in the pool. The `zpool status` command reports the progress of the remove operation until the resilvering completes.

See Oracle Solaris ZFS Device Removal, "Removing Devices From a Storage Pool" in *Managing ZFS File Systems in Oracle Solaris 11.4*, and the `zpool`(8) man page for information about removing top-level data devices.

ZFS Scheduled Scrub

By default, ZFS pool scrub runs in the background every 30 days with an automatically tuned priority. The priority of the scrub is low by default but is automatically increased if the system is idle. Scrub priority is adjusted based on the specified scrub interval, the progress, and the system load. The start time of the last successful scrub is reported by the `zpool status` command.

You can customize the scheduling of the pool scrub, including disabling it, by setting the `scrubinterval` property. See "Scheduled Data Scrubbing" in *Managing ZFS File Systems in Oracle Solaris 11.4* and see the `zpool`(8) man page for information about the `scrubinterval` and `lastscrub` properties.

Fast ZFS Based File Copying

The `reflink()` and `reflinkat()` functions enable you to copy files very quickly using the underlying ZFS technology. The `reflink()` function creates a new file with the content of an existing file without reading or writing the underlying data blocks. The existing file and the file to be created must be in the same ZFS pool.

For more information, see the `reflink`(3C) man page.

The `-z` option (fast copy) of the `cp` command uses `reflink`. See the `cp`(1) man page.

ZFS Raw Send Streams

In Oracle Solaris 11.4, you can optimize ZFS `send` stream transmissions of compressed file systems and reduce network transmission traffic by using raw ZFS `send` streams.

In previous releases, a `send` stream of a compressed ZFS file system was first decompressed upon transmission, and then, the blocks were recompressed if compression was enabled on the receiving end. In the Oracle Solaris 11.4 release, these two steps are avoided because the compressed file system blocks in the stream remain compressed. Using this optimization also reduces network transmission traffic. You can optimize a ZFS `send` stream by sending it in raw mode with the new `zfs send -w` option. Enabling this new option allows the `send` stream to encode the presence of raw blocks so that a receiving system knows to process raw blocks without compressing them.

For example, to create a ZFS file system with compression enabled and send the snapshot stream with and without the `-w` option and review the resulting stream sizes:

```
# zfs create compression=on pond/cdata
# cp -r somefiles /pond/data
# zfs snapshot pond/cdata@snap1
# zfs get compressratio pond/cdata@snap1
NAME              PROPERTY       VALUE  SOURCE
pond/cdata@snap1  compressratio  1.79x  -

# zfs send pond/cdata@snap1 > /tmp/stream
# zfs send -w compress pond/cdata@snap1 > /tmp/cstream
# ls -lh /tmp/*stream*
-rw-r--r--   1 root     root        126M Feb 15 14:35 /tmp/cstream
-rw-r--r--   1 root     root        219M Feb 15 14:35 /tmp/stream
```

Systems that run previous versions of Oracle Solaris are not able to receive such streams, and an error message will be generated.

For more information, see *Managing ZFS File Systems in Oracle Solaris 11.4*.

Resumable ZFS Send Streams

In Oracle Solaris 11.4, if a network transmission is interrupted or an error occurs, ZFS `send` streams can be restarted where they left off.

Using ZFS `send` and `receive` to transfer ZFS snapshots across systems is a convenient way to replicate ZFS file system data previously suffering from the following issues:

- A `ZFS` send operation could take many hours or days to complete. During that time, the `send` operation could be interrupted by a network outage or a system failure.
- If the `send` operation fails to complete, even if it is almost complete, it must be restarted from the beginning.
- The ZFS `send` operation might be unable to transfer large streams in the window between interruptions.
- A ZFS `recv` operation might be unable to detect and report transmission errors until the entire stream has been processed.

This Oracle Solaris 11.4 release provides a way for ZFS send streams to be resumed at the point they were interrupted with the following new options:

- `zfs receive -C` – Writes a receive checkpoint to `stdout`.
- `zfs send -C` – Reads a receive checkpoint from `stdin`.

- `zfs send -s` (nocheck) – Disables the new on-the-wire format.
- `zfs list -I` (state) – Recursively displays incomplete datasets as incomplete datasets are not displayed by default.

For more information, see *Managing ZFS File Systems in Oracle Solaris 11.4*.

Configurable ZFS Read and Write Throughput Limits

The Oracle Solaris 11.4 release provides the ability to limit a ZFS file system's reads and writes to disk. You can enable a `read` or `write` limit on a ZFS file system by setting the `readlimit` and `writelimit` properties, in units of bytes per second. Using these features allow you to optimize ZFS I/O resources in a multitenant environment.

The `defaultwritelimit` and `defaultreadlimit` properties are added to increase the manageability of a large number of ZFS file systems. If `defaultwritelimit` and `defaultreadlimit` properties are set, all file system descendants inherit the assigned value. If you apply the default `read` or `write` limit to ZFS file system, it is only applied to descendant file systems, and not to the file system itself. The read-only `effectivereadlimit` and `effectivewritelimit` properties are added to provide a view of what the effective limit is on a file system. The reported effective limit is the lowest data limit at any point between the parent and the indicated file system.

For example, you would set `read` and `write` limits as follows:

```
# zfs set writelimit=500mb pond/apps/web
# zfs set readlimit=200mb pond/apps/logdata
```

The following example shows how to display `read` and `write` limits:

```
# zfs get -r writelimit,readlimit pond/apps
NAME                PROPERTY    VALUE    SOURCE
pond/apps           writelimit  default  default
pond/apps           readlimit   default  default
pond/apps/logdata   writelimit  default  default
pond/apps/logdata   readlimit   200M     local
pond/apps/web       writelimit  500M     local
pond/apps/web       readlimit   default  default
pond/apps/web/tier1 writelimit  default  default
pond/apps/web/tier1 readlimit   default  default
```

You can display the effective write limit as follows:

```
# zfs get effectivewritelimit pond/apps/web
NAME           PROPERTY             VALUE  SOURCE
pond/apps/web  effectivewritelimit  500M   local
```

For more information, see *Managing ZFS File Systems in Oracle Solaris 11.4*.

Monitor and Manage ZFS Shadow Migration

The Oracle Solaris 11.4 release provides improved ZFS shadow migration operation with enhancements for better visibility in monitoring migration errors and controlling in-progress migrations. The following new options are introduced:

- `shadowstat -E` and `-e` – Provided for monitoring migration errors for all migrations or a single migration.

- `shadowadm` – Control in-progress migrations.

For example, you can identify shadow migration errors of multiple migration operations:

```
# shadowstat
                                    EST
                          BYTES     BYTES           ELAPSED
DATASET                   XFRD      LEFT    ERRORS  TIME
tank/logarchive           16.4M     195M    1       00:01:20
pond/dbarchive            4.49M     248M    -       00:00:51
tank/logarchive           16.6M     194M    1       00:01:21
pond/dbarchive            4.66M     248M    -       00:00:52
tank/logarchive           16.7M     194M    1       00:01:22
pond/dbarchive            4.80M     248M    -       00:00:53
tank/logarchive           17.1M     194M    1       00:01:23
pond/dbarchive            5.00M     248M    -       00:00:54
tank/logarchive           17.3M     194M    1       00:01:24
pond/dbarchive            5.16M     247M    -       00:00:55
```

You can identify the specific migration error as follows:

```
# shadowstat -E
tank/logarchive:
PATH                                            ERROR
e-dir/socket                                    Operation not supported
pond/dbarchive:
No errors encountered.
```

For example, to cancel the migration as the open socket cannot be migrated:

```
# shadowadm cancel tank/logarchive
```

For more information, see *Managing ZFS File Systems in Oracle Solaris 11.4*.

Preserving ZFS ACL Inheritance

In Oracle Solaris 11.4 , a new ZFS ACL feature helps in providing a better experience when sharing a ZFS file system over both the NFS and Server Message Block (SMB) protocols. A new inheritance value for the `aclinherit` property is introduced, which allows passthrough semantics but overrides the permissions set in the inherited `owner@`, `group@`, and `everyone@` ACEs with the values requested in the `open`, `create`, or `mkdir` system call. When set, any inheritable ACEs have their inherit bits preserved. This behavior is important to allow SMB and NFS sharing to inherit ACLs in a natural way. The new value is called `passthrough-mode-preserve`. No changes to the `aclmode` property occurred, but a `chmod` operation takes into account what the inheritance behavior is with respect to the `aclinherit` property. In particular, it preserves the inheritance bits during a `chmod` operation.

For more information, see *Managing ZFS File Systems in Oracle Solaris 11.4*.

NFS Version 4.1 Server Support

Oracle Solaris 11.4 includes server support for NFS version 4.1. The protocol provides the following new features and considerations:

- Exactly Once Semantics (EOS) – Provides a reliable duplicate request cache for the NFS version 4.1 protocol. This duplicate request cache guarantees that non-idempotent requests such as the remove request executes only once, even in cases of transient network failures and retransmissions. This feature eliminates the long-standing problems with NFS version 3 and NFS version 4.

- `reclaim_complete` – A new protocol feature that enables the server to resume NFS service quickly after a server restart. Unlike NFS version 4, the user does not need to wait for a specific amount of time, known as the grace period, before returning to service. With `reclaim_complete`, the server can end the grace period once all clients have recovered. This feature is particularly important for high-availability environments.
- Planned GRACE-less Recovery (PGR) – Allows an Oracle Solaris NFS version 4 or NFS version 4.1 server to preserve the NFS version 4 state across NFS service restarts or graceful system reboot, so that the NFS version 4 server does not enter the GRACE period to recover NFS version 4 state. The advantage is that NFS client applications can avoid a data downtime of potentially 90 seconds across NFS service restarts and graceful system reboots.
- Consider the following interoperability issues:
  - Oracle Solaris NFS version 4.1 server supports both Linux clients and VMware. However, delegation should be disabled on the server for Linux clients.
  - There are known issues with locking and in-state recovery when using delegation with the Linux client.

    ```
    # sharectl set -p server_delegation=off nfs
    ```

    You can disable NFS version 4.1 support on the server as follows:

    ```
    # sharectl set -p server_versmax=4.0 nfs
    ```

For more information, see *Managing Network File Systems in Oracle Solaris 11.4*.

NFSv3 Mount Using TCP

When a file system is mounted using NFS Version 3 and TCP is the selected transport, the initial mount setup will also use TCP as the transport. In previous Oracle Solaris releases, UDP would be used for the mount setup and TCP would be used only after the mount had been established.

If you allow NFS mounts through a firewall, this feature might enable you to simplify your firewall configuration.

This feature also enables you to use NFS Version 3 at sites where UDP traffic is blocked.

For more information, see *Managing Network File Systems in Oracle Solaris 11.4* and the `mount_nfs`(8) man page.

Extended File System Attributes in `tmpfs`

`tmpfs` file systems support extended system attributes. See the `tmpfs`(4FS) and `fgetattr`(3C) man pages.

SMB 3.1.1 Support

Oracle Solaris 11.4 provides SMB 3.1.1 protocol support in the Oracle Solaris SMB server, which includes the following SMB features:

- Continuously Available Shares – This feature enables an Oracle Solaris SMB server to make shares continuously available in the event of a server crash or reboot.
- Multichannel – This feature enables an Oracle Solaris SMB file server to use multiple network connections per SMB session to provide increased throughput and fault tolerance.

- Encryption – This feature enables an Oracle Solaris SMB server to encrypt SMB network traffic between clients and the server. SMB encryption secures SMB sessions and protects against tampering and eavesdropping attacks.

For more information, see *Managing SMB File Sharing and Windows Interoperability in Oracle Solaris 11.4* and the `smbstat`(8) man page.

Networking Features

This section describes the networking features that are new in this release. These features enhance the existing networking technology and software defined networking to build services that meet organizational performance requirements, and to provide greater application agility and the flexibility you demand.

Migration of Persistent Network Configuration to SMF

In Oracle Solaris 11.4, persistent network configuration has been migrated to the Service Management Facility (SMF) repository. This move enables you to customize networking parameters and create more complex network configurations during an automated installation. This change also aligns network configuration with other system components that use SMF as a core storage repository. To accommodate the migration of network configuration parameters to SMF, certain `dladm` property names have also changed.

For more information about the `dladm` property name changes, see Chapter 2, "Administering Datalink Configuration in Oracle Solaris" in *Configuring and Managing Network Components in Oracle Solaris 11.4*, and the `dladm`(8) man page. For more information, see *Automatically Installing Oracle Solaris 11.4 Systems*.

Oracle Solaris Client Side Support for IEEE 802.1X

In Oracle Solaris 11.4, you can authenticate Oracle Solaris clients using the IEEE802.1X standard. Previously, if a secure LAN that was deployed needed client systems to be authenticated prior to providing any services, the authentication was not supported on an Oracle Solaris system.

This new feature allows network administrators to configure an Oracle Solaris client system, and enables it to be authenticated by a server behind a secured LAN. The Oracle Solaris system has the `nacd` daemon running. The client system is connected to a secured LAN through a port on a switch on the LAN. The daemon communicates with a server on the secured LAN to get authenticated, before it can get a service such as DHCP from the LAN. A network administrator can also use the `nacadm` command to configure the security credentials, and use the `dladm` command to enable or disable the authentication on a given link.

For more information, see the `nacd`(8), `nacadm`(8), and `dladm`(8) man pages. You can also see .

Datacenter TCP

Oracle Solaris 11.4 includes Datacenter TCP (DCTCP), an improvement to TCP congestion control for datacenter traffic. DCTCP uses improved Explicit Congestion Notification (ECN) processing to estimate the fraction of bytes that encounter congestion, rather than simply detecting that congestion has occurred. DCTCP then scales the TCP congestion window based on this estimate. This method achieves high burst tolerance, low latency, and high throughput with shallow-buffered switches.

For more information, see "Implementing Traffic Congestion Control" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Live Migration and HA Failover for SR-IOV Enabled `anets`

Oracle Solaris 11.4 delivers the following new DLMP functionality and relevant support for Oracle Solaris kernel zones:

- A novel DLMP architecture that enables the creation of SR-IOV VNICs and IPoIB partitions in the global zone
- Live migration support for SR-IOV enabled `anets` in Oracle Solaris kernel zones
- High Availability failover for SR-IOV enabled `anets` in Oracle Solaris kernel zones

For more information, see Chapter 2, "Configuring High Availability by Using Link Aggregations" in *Managing Network Datalinks in Oracle Solaris 11.4*.

Specifying a Name for a Persistent Static Route

In Oracle Solaris 11.4, the `route` command adds a `-name` option to enable the naming of persistent static routes. This route name can be used to change, get, or delete the static route. The name can also be used to distinguish between different persistent static routes.

For more information, see the `route(8)` man page. You can also see "How to Specify a Name for a Persistent Route" in *Configuring and Managing Network Components in Oracle Solaris 11.4*.

Open Fabric Enterprise Distribution 3.18

The following components of the InfiniBand (IB) OS-bypass framework from the Open Fabrics Enterprise Distribution (OFED) have been updated to version 3.18:

- InfiniBand verbs (transport) library
- Userspace driver for Mellanox ConnectX InfiniBand HCAs
- Userspace RDMA communication manager (CM) library
- InfiniBand management datagram (MAD) library
- InfiniBand userspace management datagram (uMAD) library
- User level InfiniBand management utilities
- InfiniBand diagnostic tools
- `qperf` command

For more information, see the man pages delivered with the `network/open-fabrics` packages.

ISC BIND 9.10.3

Oracle Solaris 11.4 includes an updated version of ISC BIND that provides many new features such as automatic resigning of DNSSEC zones, new statistics reporting, and response late limiting. Users can now deploy DNSSEC easily. Monitoring is simplified and the service is more robust against DOS attacks.

Performance and Observability

This section describes the platform and performance enhancements that are new in this release. These features help optimize Oracle Solaris for SPARC and x86 based systems, thereby increasing performance and provide better diagnosis for your systems.

DTrace SCSI Provider

The Oracle Solaris 11.4 release introduces a new DTrace SCSI provider that is designed to trace SCSI commands and task management functions that are issued by an Oracle Solaris system. The SCSI provider has the following benefits:

- Enables you to trace SCSI commands on an Oracle Solaris system without knowing the internal structure
- Includes probes and structures that follow the SCSI T10 standards as much as possible
- Provides a counter-part to the DTrace I/O provider that traces I/O traffic at a different layer
- Delivers a `scsitrace` script that consumes the new probes

The following example illustrates a one-line trace that identifies SCSI target resets:

```
# dtrace -n 'scsi:::tmf-request
                      /(args[1] == SCSI_TMF_TARGET_RESET) &&
                       (args[0]->addr_path != "NULL")/ {
        printf("Target Reset sent to %s", args[0]->addr_path);}'
```

For more information, see the "iscsi Provider" in *Oracle Solaris 11.4 DTrace (Dynamic Tracing) Guide*.

DTrace `fileops` provider

The `fileops` provider exposes a full set of standard UNIX file operation probes that are intended more for an Oracle Solaris administrator than for a developer. For example, the provider can display `read` or `write` latency information for all file systems, including pseudo file systems.

The `fileops` probes pertain to the file operations: `open`, `close`, `read`, `write` and so on. These probes are neither specific to any file system type, nor are they dependent on I/O to external storage devices. For example, the `fileops:::read` probe fires on any read from a file, regardless of whether the data comes from disk or is cached in memory.

You can use the `read` probe to observe read latencies on different file system types. For example:

```
fileops:::read
{
    @[args[0]->fi_fs] =
    quantize(args[1]);
}
```

The resulting output provides a graph of read counts and latencies across all file system types on the system.

For more information, see the "fileops Provider" in *Oracle Solaris 11.4 DTrace (Dynamic Tracing) Guide*.

DTrace MIB Provider for TCP, UDP, and IP

The Oracle Solaris 11.4 release extends the existing DTrace MIB provider for observing events in the networking stack with protocol information so that TCP, UDP, and IP connections can be identified.

For more information, see the "mib Provider" in *Oracle Solaris 11.4 DTrace (Dynamic Tracing) Guide*.

DTrace `pcap()` Action

A new action, `pcap()`, is added to DTrace. The `pcap()` action will do one of the following:

- Display packet data as `tracemem()` does, but coalesced into a contiguous buffer.
- If `freopen()` has specified a capture file, the `pcap()` action will capture packet data to a packet capture file via the `libpcap` function `pcap_dump()`. DTrace does the following with the packet data:

  1. Collects packet data in probe context.
  2. Coalesces the packet data into a contiguous buffer if the data is not already in a contiguous buffer.
  3. Dumps the data to the specified file via the `pcap_dump()` function, which was called when collecting the data.

The following `pcap()` action dumps memory to `stdout` as `tracemem()` does:

```
pcap(mblk, protocol);
```

The following calls dump packet data to the capture file with a suffix that specifies the current `pid`:

```
freopen("/tmp/cap.%d", pid);
pcap(mblk, protocol);
```

This allows you to collate packet traces by process or service, for example. Because `freopen()` is classified as a destructive action, the above script must specify the `-w` (”destructive”) `dtrace` option. The `pcap()` action is not destructive.

DTrace `print()` Action

DTrace has a new `print()` action to display arbitrary types, as shown in the following example:

```
# dtrace -q -n 'fop_close:entry {print(*args[0]);exit(0)}'

vnode_t {
 v_lock = {
   _opaque = [ NULL ]
 }
 v_flag = 0x0
 v_count = 0x1
 v_data = 0xffffc10054425378
 v_vfsp = specfs`spec_vfs
 v_stream = 0xffffc100623354e8
 v_type = VCHR
 v_rdev = 0xee00000026
 v_vfsmountedhere = NULL
 v_op = 0xffffc10029d98040
 v_pages = NULL
 v_filocks = NULL
 v_shrlocks = NULL
 v_nbllock = {
   _opaque = [ NULL ]
 }
 v_cv = {
   _opaque = 0x0
 }
 v_pad = 0xbadd
 v_count_dnlc = 0x0
 v_locality = NULL
 v_femhead = NULL
 v_path = "/devices/pseudo/udp@0:udp"
 v_rdcnt = 0x0
 v_wrcnt = 0x0
 v_mmap_read = 0x0
 v_mmap_write = 0x0
```

```
 v_mpssdata = NULL
 v_fopdata = NULL
 v_vsd_lock = {
   _opaque = [ NULL ]
 }
 v_vsd = NULL
 v_xattrdir = NULL
 v_fw = 0xbaddcafebaddcafe
}

#
```

For more information, see "print Action" in *Oracle Solaris 11.4 DTrace (Dynamic Tracing) Guide*.

Kstats v2 Framework

The Kernel statistics (kstats) v2 framework provides better performance and a number of optimizations, when compared to the previous kstat framework. Some of the new notable components included are:

- Kernel API which provides the functionality to create and manipulate v2 kstats. Kstats are identified using a unique URI and include metadata for both the kstat and the name-value pairs that the kstat contains. This API allows the kstat to describe the values that it is reporting.
- `libkstat2` library that which provides access to v2 kstats created in the kernel. Kstats are looked up through their unique URI and are presented as hashmaps. Developers can subscribe to events at a particular kstat URI level and will be notified when any kstats under them in the URI tree are added or removed below them in the URI tree.
- `/usr/bin/kstat2` utility which provides CLI access to kstat. This new utility examines the available kstats on the system and reports those statistics that match the criteria specified on the command line. Each matching statistic is then printed with its URI and its actual value. A number of different output formats are supported, including human-readable, parseable and JavaScript Object Notation (JSON) formats.

For information about the kernel API, see the `kstat2_create`(9F), `kstat2_create_with_template`(9F), and `kstat2_create_histogram`(9F) man pages. For information about the `libkstat2` library, see the `libkstat2`(3LIB) and `kstat2`(3KSTAT2) man pages. For information about the `kstat2` utility, see the `kstat2`(8) man page.

FMA Core File Diagnostics

Oracle Solaris 11.4 includes the core file diagnostics feature, which provides a summary of basic telemetrics from userland core files, raises FMA alerts to notify the user, and provides a diagnostic core retention policy and SMF case association.

The diagnostic core files contain only necessary content, which makes the size of the content small. The core files will be deleted after the text summary files are generated thereby, reducing disk space. With other new features such as stack diagnosis, FMA will be able to search the stacks in the summary file in the Oracle database for known issues. The retention policy enables the user to set the diagnostic core policy through the `coreadm` command. The `coreadm` command also provides functionalities such as deleting the cores immediately or keeping a certain number of cores for a certain time. The case association feature is for the `sw-diag-response` diagnosis engine. All core diagnosis alerts leading to a software service failure can be viewed together along with each event's set of stack and environment data.

The user now has more control over diagnostic cores. When a software service does not run properly and is taken out of service, the administrator can easily and quickly view all the events that led to the service

failure and be better informed about the processes that failed and where each process failed in its code execution.

For more information, see the coreadm(8) man page.

`pfiles` Enhancements

In Oracle Solaris 11.4, the `pfiles` command accepts a core file name as an argument and can now display information about file descriptors opened by a process which dumps core. This functionality provides additional help in debugging the process core dump for the root cause of the dump.

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

Monitor I/O Latency via `fsstat`

The `fsstat` command has a new `-l` option that reports latency information for `read`, `write`, and `readdir` operations. The latency information is independent of physical I/O operations, and therefore is representative of file system performance, as seen by applications. This feature enables users to observe file system latency for file system types or individual file systems. This feature is useful in troubleshooting file system performance problems.

For more information, see the fsstat(8) man page.

SCSI I/O Response Time Distribution Statistics

Oracle Solaris 11.4 now provides SCSI I/O response time or I/O latency distribution information for better observability. I/O response time distribution can be used to identify response time outliers. The distribution is stored in a histogram with three different x-scale options: `linear`, `log2-based`, and `log10-based`. The distribution can be displayed using the `iostat` command. The `-L` option is added in conjunction with the `-x` and `-Y` options to show the histogram. This distribution information can be used to investigate performance issues.

For more information, see the sd(4D) and iostat(8) man pages.

Virtualization Features

This section describes virtualization features that are new in this release. These features provide efficient cloud virtualization with no loss in performance and enable you to run large scale applications in the cloud with the optimized use of resources.

Configure Immutable Zones by Running in the Trusted Path

The immutable file system feature that was first introduced in Oracle Solaris 11 11/11 (read-only root for nonglobal zones) has been significantly extended so that immutable zones are now much easier to adopt and use.

Previously, to make certain configuration changes in immutable zones, you had to make the zone temporarily mutable. In Oracle Solaris 11.4, you can run in the Trusted Path Domain (TPD) while the zone is still immutable for other users.

To run in the TPD, do one of the following:

- Add the user to the `/etc/security/tpdusers` file and set `start/trusted_path` to `true` on the `console-login` service.
- For remote RAD access to the Trusted Path, set `method_context/trusted_path` to `true` on the `rad:remote` service and add `tpd=yes` to the `user_attr` entry for each user that is allowed remote access to the TPD.

These procedures are described in detail in "Administering an Immutable Zone by Using the Trusted Path Domain" in *Creating and Using Oracle Solaris Zones*.

In addition to administrators running in the TPD, you can configure some services to run in the TPD as described in "SMF Services in Immutable Zones" in *Creating and Using Oracle Solaris Zones*.

Silicon Secured Memory Support in Oracle Solaris Kernel Zones

Software in Silicon Support in Oracle Solaris kernel zones is enhanced to include Silicon Secured Memory (SSM). SSM adds real-time checking of access to data in memory to help protect against malicious intrusion and flawed program code in production for greater security and reliability.

SSM protection is utilized by Oracle Database 12c by default, and is easy to turn on for other applications. See "Software in Silicon Features on Kernel Zones" in *Creating and Using Oracle Solaris Kernel Zones*.

See also "Silicon Secured Memory Security Exploit Mitigations" on page 5.

Oracle Solaris Kernel Zone Support for SPARC M7 and M8 DAX Coprocessors

This feature enables Oracle Solaris software to make use of the SPARC M7 and M8 Data Analytics Accelerator (DAX) query functionality by using the High Performance Kernels (HPK) library of the Oracle in-memory database product, when running on an Oracle Solaris kernel zone.

The HPK library in the RDBMS product provides hardware-optimized operations on vector or columnar data for In-Memory Columnar (IMC) databases. The library uses hardware-specific capabilities to efficiently perform operations, and can make use of the DAX query capabilities when available in a Oracle Solaris kernel zone.

For more information, see *Oracle Solaris Zones Configuration Resources*, "Software in Silicon Features on Kernel Zones" in *Creating and Using Oracle Solaris Kernel Zones*, and the `zonecfg(8)` man page.

VLAN Aware Oracle Solaris Kernel Zones

VLANs split a single Layer 2 (L2) network into multiple logical networks so that each logical network is its own broadcast domain. This feature means that all the devices connected to a VLAN can see the broadcast frame of every other device irrespective of its physical location.

Previously, Oracle Solaris kernel zones were able to assert only one VLAN ID. In the Oracle Solaris 11.4 release, you are able to assert additional VLAN IDs per `anet` by specifying the new `vlan` resource type in a zone configuration.

For more information, see "About Configuring Virtual LANs in Kernel Zones" in *Creating and Using Oracle Solaris Kernel Zones*.

Delegated Oracle Solaris Zones Restarter

The zones delegated restarter (`system/zones:default`) enables control of boot order using dependencies and priorities. In previous Oracle Solaris releases, this service did not provide a way to prioritize and manage zone boot order. If applications in different zones on the same system depend on each other, for example, you might want to be able to boot all those zones in parallel.

In addition to providing milestones for zone booting, the zones delegated restarter also provides the ability to add dependencies for other zones or other services. For example, `Zone-C` can be configured to start after `Zone-A` and `Zone-B` have finished booting, or after a firewall service has started.

See the `svc.zones(8)` and `zonecfg(8)` man pages for information about the new `boot-priority` and `smf-dependency` properties.

Live Zone Reconfiguration for Datasets on Oracle Solaris Native Zones

The ability to change the configuration of an Oracle Solaris zone without causing an outage to the end user or service is key to meeting present day service level agreements (SLAs). With Live Zone Reconfiguration (LZR), users are able to make changes to Oracle Solaris zone configuration and push them to a running zone either as a permanent or a temporary change, without the need to reboot the zone.

In the Oracle Solaris 11.4 release, you are able to add or remove ZFS datasets to and from an Oracle Solaris native zone by using the LZR methodology.

For more information, see Chapter 4, "Live Zone Reconfiguration of Kernel Zones" in *Creating and Using Oracle Solaris Kernel Zones*.

Moving Oracle Solaris Zones

Oracle Solaris 11.4 enables you to use the `zoneadm` command with the `move` subcommand to move an installed Oracle Solaris zone across different storage URIs. You can perform the following actions:

- Move an Oracle Solaris zone from a local file system (default) to shared storage
- Move an Oracle Solaris zone from shared storage to a local file system
- Move an Oracle Solaris zone from one shared storage location to another, while also changing the `zonepath`
- Change the `zonepath` without moving the Oracle Solaris zone installation

For more information, see the `solaris(7)`, `zones(7)`, and `zoneadm(8)` man pages. You can also see *Creating and Using Oracle Solaris Zones*.

Zone Cold Migration

In Oracle Solaris 11.4, Oracle Solaris zones in an installed state using shared storage can be migrated to another system using the `zoneadm` command. Oracle Solaris kernel zones in an installed or suspended state using shared storage can also be migrated. Non-running Oracle Solaris zones and Oracle Solaris kernel zones can be evacuated using the `sysadm` command allowing enhanced zone availability during global zone scheduled downtimes.

For more information, see `zoneadm(8)`, `solaris(7)`, `solaris-kz(7)`, and `sysadm(8)` man pages.

Multipathing for Oracle VM Server for SPARC Virtual SCSI HBAs

In Oracle Solaris 11.4, the Virtual HBA subsystem supports physical HBA drivers that have multipathing enabled in their respective `driver.conf` files. This feature allows Oracle Solaris I/O Multipathing to be supported in a `sun4v` Service Domain on a per-HBA port basis, as described in *Managing SAN Devices and I/O Multipathing in Oracle Solaris 11.4*.

The `vhba` module has supported multipathing in the Guest Domain since the initial release of Oracle Solaris 11.3. Allowing multipathing in both domains of a `sun4v` system improves fault tolerance and I/O throughput to SCSI devices.

Device Masking for Oracle VM Server for SPARC Virtual SCSI HBAs

Oracle Solaris 11.4 allows a user to configure a virtual SAN (Storage Area Network) device so that it represents an explicit set of physical SCSI devices. The original and default behavior of a vSAN device is to represent *all* physical SCSI devices that are reachable from the user-specified SCSI HBA Initiator port.

In Oracle Solaris 11.4, a user can enter commands to dynamically add and remove explicit physical SCSI devices from a specified vSAN device. By associating a specific vSAN device with a specific Guest Domain, the user has complete control over which physical SCSI devices can be accessed by a specific Guest Domain.

System Management Features

This section describes the system management features that are new in this release. These features enable you to configure services with seamless configuration management, automate configuration enforcements across systems, and provide secure, remote administrative access.

See also "Configure Immutable Zones by Running in the Trusted Path" on page 18.

System Data Visualization and Performance Analysis With Oracle Solaris Web Dashboard

Oracle Solaris provides many system observability tools, including statistics tools (such as `mpstat`), DTrace, and audit records. The Analytics application of the Oracle Solaris Web Dashboard enables you to more easily view different kinds of single system and application performance data together graphically for more effective observability and analysis. Identify current system state, and visualize statistics, faults, and administrative changes over time and compared with other time periods. Open a data sheet to see more details and additional related data. Compare current and historical data graphs to visualize anomalies and trends, view related data in graphs in additional charts on the same page, and view events such as audit events on the same graph with other data.

The Web Dashboard is a web application that visually shows current and recent performance, any recent system faults, and other events. This combination of information helps you determine which system resources to examine to diagnose problems.

You can view the average utilization for all of a selected type of resource or view the utilization of a single resource. For example, you can determine which applications are responsible for most of the network traffic on a system. Similarly, in addition to overall CPU usage data, you can select the zone partition of CPU

usage data to determine which zones are using the CPU the most. You can determine whether a resource is allocated to a particular zone or workload.

For more information, see *Using Oracle Solaris 11.4 Analytics*. For information about how to store your own data in the underlying statistics store, see *Adding Custom Data to Oracle Solaris 11.4 Analytics*.

Displaying DAX Utilization and Performance

On SPARC M7, M8, T7, and T8 systems, you can use the `daxstat` command to display DAX statistics (per-dax, per-cpu, or per-queue) in tabular form.

The `daxinfo` command enables users to determine the static configuration of DAX hardware available on a system. This information can be used for performance and diagnostic reporting.

For more information, see "Displaying DAX Information" in *Managing System Information, Processes, and Performance in Oracle Solaris 11.4* and the `daxstat`(8) and `daxinfo`(8) man pages.

SMF Nested Property Groups

The Oracle Solaris Service Management Facility (SMF) provides an infrastructure for storing system configuration data in a central location, rather than within the application-specific configuration files. The previous SMF modeling capability was limited in its ability to model affiliations among configuration data.

The Oracle Solaris 11.4 release modifies the SMF property group relationship so that a property group cannot only be parented by a service or instance, but can also be parented by another property group. With the introduction of this relationship, the ability to model a greater variety of system configuration data is now possible. A consequence of adding this relationship is that when uniquely identifying a property group, you must not only consider the name, but also the lineage.

Nested property groups are subject to the same privilege model as property groups, as well as the same template verification as property groups. For more information, see the `smf_template`(7) man page.

New SMF Profile Layers

Oracle Solaris 11.4 introduces three new SMF profile layers: `enterprise-profile`, `node-profile`, and `sysconfig-profile`. In previous releases, SMF configuration could only be applied to the `site-profile` and `admin` layers. In this release, by using multiple layers, you can apply generally useful configuration to the `enterprise-profile` layer, location-specific configuration to the `site-profile` layer, and host-specific configuration to the `node-profile` layer. This feature enables the effective use of SMF profiles in environments where subsets of systems or individual systems need to override more general configuration.

For more information, see "Repository Layers" in *Managing System Services in Oracle Solaris 11.4* and Chapter 5, "Configuring Multiple Systems" in *Managing System Services in Oracle Solaris 11.4*. See also the `smf`(7), `svccfg`(8), and `svcprop`(1) man pages and the `sysconfig`(8), `solaris`(7), `solaris-kz`(7), and `smf_bootstrap`(7) man pages.

Goal Services

Oracle Solaris 11.4 includes a new type of services, goal services, which provides a single point of monitoring for configurable set of services, on which the goal services are dependent. If a dependency of a

goal service cannot be satisfied, the goal service enters the maintenance state and generates a software FMA alert.

For more information, see Chapter 7, “Creating a Service that Notifies if Conditions are not Satisfied” in *Developing System Services in Oracle Solaris 11.4* and the `smf`(7) and `svcadm`(8) man pages.

Diagnosing Device Hotplugging Failures

In Oracle Solaris 11.4, the new Device Usage Information feature helps in diagnosing `hotplug` failures. In previous releases, when the `hotplug` command failed to remove a busy device, the error message “Devices or resources are busy.” was displayed with no further explanation, which made it difficult to diagnose the problem. Now with this feature, additional information is displayed explaining what opened or held the device, which helps in diagnosing the problem.

For example, to check if the device is busy, use the following command:

```
# hotplug offline /pci@0,0 pci.1,0
ERROR: devices or resources are busy.
/pci@0,0/pci8086,3408@1/pci1000,1000@0/sd@0,0:
   { Hold: module devfs (modid=6). }
   { Hold: module specfs (modid=3): spec_open() }
   { Open: process format[123501]. }
   { Open: module zfs (modid=49). }
```

For more information, see *Managing Devices in Oracle Solaris 11.4*.

`sysadm` Utility

Oracle Solaris 11.4 introduces the new `sysadm` utility that can be used to control maintenance mode for a system, and perform evacuation of zones hosted on the system. Starting a maintenance mode logs an audit record, and prevents the subsequent attach, boot, or incoming migration of any zones into the system. The host may be evacuated by migrating all `solaris-kz` brand zones away from the system to targets specified in a zones SMF service instance.

Maintenance mode and evacuation can be used to perform administration on a zones host, or remove it from service. You can prevent new zones from attaching or booting, migrate away any running kernel zones, perform some maintenance requiring a reboot such as updating Oracle Solaris, and finally migrate back those kernel zones, with just a few commands and no disruptions to the running kernel zones.

For more information, see the `sysadm`(8) man page.

Automating OpenLDAP and OUD Server Configuration

The `ldapservercfg` utility automates the configuration of the OpenLDAP server and Oracle Unified Directory (OUD) server to support Oracle Solaris LDAP naming services and account management. The utility is integrated with the `svc:/network/ldap/server:openldap` SMF service and automatically configures the server when it is first enabled. It can also be run interactively to customize the OpenLDAP server configuration or to configure the OUD server. The `ldapservercfg` utility helps users to easily deploy OpenLDAP and OUD server on Oracle Solaris systems. It also enables the functionality for remote account management using RAD.

For more information, see the `ldapservercfg(8)` man page.

Puppet Configuration Management Software

Puppet is cross-platform software that you can use to automate the configuration management of multiple platforms, including Oracle Solaris servers and their subsystems.

You can use Puppet to standardize and enforce resource configurations across your entire IT infrastructure. Oracle Solaris 11.4 includes core Puppet resource types, including files, packages, users, and services. In addition, many modules are included for managing other third-party software on Oracle Solaris. Finally, several Oracle Solaris specific resource types are provided for use in the Oracle Solaris release, such as Oracle Solaris zones.

The new Oracle Solaris Puppet configuration option `config/degrade_smf_on_error` `Degrade_smf_on_error` causes the `puppet:agent` service to change state to degraded when a resource error occurs during Puppet manifest application. Puppet continues to run after being marked degraded. This option makes resource errors in Puppet manifest applications more visible to the user.

Note that Oracle Solaris 11.4 supports Puppet 5.5. This software package is not installed by default on your system.

If a system has a previous Puppet version installed, that version will be automatically upgraded to Puppet 5.5. See "What's New in Puppet in Oracle Solaris 11.4" in *Using Puppet to Perform Configuration Management in Oracle Solaris 11.4* for important information about this upgrade.

For general information about Puppet, refer to Puppet Documentation.

MCollective

The Marionette Collective, known as MCollective is a framework that enables you to more easily build and manage a large number of servers. While working with a large number of servers, it is difficult to rely on a static list of tools for system management. MCollective uses a discovery method that is based on metadata, as well as filtering to search for hosts.

MCollective also uses a publish-subscribe pattern to broadcast request to all of the servers that are connected to the middleware component. These requests have filters that are attached so that only those servers matching a filter will act on the requests.

For more information, see *Using Puppet to Perform Configuration Management in Oracle Solaris 11.4* and Using MCollective Command Line Applications.

Augeas

Augeas is a library and command-line tool that provides a unified way to edit UNIX configuration files of differing formats. When invoked, Augeas provides a command-line interface where configuration files can be read and presented in a tree format. This data can then be manipulated within the Augeas interface. Data is then translated back into the format of the original configuration files for saving.

Augeas provides a single public API for manipulating various UNIX configuration files. Other applications can leverage this API, rather than having each application provide its own solution for modifying configuration files.

Default User Attributes for LDAP Accounts

In Oracle Solaris 11.4, the functionality of qualified user attributes is extended to provide default settings for specific hosts or netgroups. LDAP-based user accounts without explicitly assigned attributes can inherit

the default attributes for the host on which they are executed. If no host-based default attributes is specified, and the user is a member of a netgroup, then the attributes associated with the netgroup are inherited. This functionality simplifies administration of LDAP-based accounts by enabling them to share common user attributes based on netgroup membership.

For more information, see the `user_attr`(5), `useradd`(8), `userdel`(8), and `usermod`(8) man pages.

`useradm` Tool

The `useradm` tool is an interactive menu-based tool for administration of user and role accounts. This tool is a replacement for the Visual Panels User Manager. The tool is implemented as a RAD client using the Python bindings described in `usermgr(3rad)`, and can be run on any terminal window. The user interface consists of hierarchical menus containing lists of available selections.

The menu-based interface of `useradm` simplifies the management of users and roles, supporting all aspects of account management such as access rights, auditing, and password management. All valid choices are presented in scrolling lists that enable you to interact with the keyboard and make selections by using the appropriate keystroke.

For more information, see the `useradm`(8) man page.

Fault Output Identifies Bugs

A new `stackdiag` feature enables `fmadm faulty` to display a list of bugs that might have caused the fault. Using this known bug list, you can look for solutions on My Oracle Support.

For more information, see *Managing Faults, Defects, and Alerts in Oracle Solaris 11.4* and the `stackdiag`(1) man page.

`fcinfo` Utility

In Oracle Solaris 11.4, the `fcinfo` utility has been enhanced to provide the following functionality:

- Establish whether an HBA port has access to a remote port or not. The `fcinfo remote-port` command can show the number and details of all remote ports for each HBA port when no option is specified.
- Establish the path class and the path state for all LUNs presented by an individual storage array. The `fcinfo lu -v` command can specify the remote port and node World Wide Name (WWN) with the `-P` and `-N` options respectively.

For more information, see the `fcinfo`(8) man page.

New IPS Repository Management Options

The Oracle Solaris Image Packaging System (IPS) has new options to help manage package repository access and troubleshoot some installation and update issues:

- Ignore network cache. A new global option named `--no-network-cache` is added to ignore cached network data.

  Problems with accessing package data can be caused by caching proxies between the package client and network-based package repositories. To troubleshoot such problems, use the `--no-network-cache`

option to always get package data from a repository and not from any caches that an HTTP proxy might have.

For more information, see "Initial Troubleshooting Steps" in *Updating Systems and Adding Software in Oracle Solaris 11.4* and the `pkg`(1) man page.

- Enable or disable a specific origin. The combination of `-d`, `-e`, and `-g` of the `pkg set-publisher` subcommand can be used to enable or disable a specific origin of a publisher.

  For more information, see "Enabling and Disabling Publisher Origins" in *Updating Systems and Adding Software in Oracle Solaris 11.4* and the `pkg`(1) man page.

IPS also has better error messaging. For example, an error message that reports a missing file entry also includes the package that delivers the file.

Installation and Software Management Features

This section describes the installation and software management features that are new in this release. These features enable fast updates and deployments through software installation and software management tools.

Starting System Services on First Boot

The `svc-create-first-boot` tool provides a single interface to create, customize, and publish a first boot service package. Provide the package repository path and first boot script as command-line arguments; the tool will publish the first boot service package to the specified repository. Using this tool simplifies automatically executing scripts at first boot.

For more information, see the `svc-create-first-boot`(1) man page.

RAD API for Automated Installer

Oracle Solaris 11.4 introduces the RAD API for Automated Installer. This API provides functionality for the remote administration of an Automated Install (AI) Server. You can write programs to manage AI server using any RAD supported client language.

For more information, see the `autoinstall(3rad)` man page.

Automated Installer Support for HMAC-SHA256

Oracle Solaris AI installations can now be secured by HMAC-SHA256. Administrators can choose HMAC-SHA256 as the policy for the HMAC algorithm for securing new AI services and clients. Administrators can also upgrade HMAC type for services and clients to be enforced immediately, or generate HMAC-SHA256 keys to be installed by the user, after which HMAC-SHA256 can be enforced by the administrator. Existing systems secured with HMAC-SHA1 will continue to be secured with HMAC-SHA1 until upgraded.

HMAC-SHA256 provides authentication and integrity for the early boot phases of an Oracle Solaris AI installation using the WAN boot protocol. This support in AI, coupled with SPARC OBP firmware support, ensures modern standards of security for wide-area network installations.

For more information, see the `installadm`(8) man page.

Dehydration and Rehydration for Oracle Solaris Unified Archives

Dehydration and Rehydration for Oracle Solaris Unified Archives (UAR) enhances the current archive technology to utilize the IPS dehydrate/rehydrate technology and minimize the footprint of a created UAR.

In this context, to dehydrate an archive means to remove all noneditable packaged files and packaged hardlinks from the alternate root of an archive image. A noneditable packaged file is a file delivered by the currently installed version of a package that has no preserve or overlay attribute, or has no tagged value of `dehydrate=False`. On the other hand, rehydration reinstalls all the files and hardlinks removed by dehydration to restore the archive image to its original state.

This feature helps to facilitate independent software vendors (ISVs) to deliver application stacks inside of a UAR where the base OS would be dehydrated thus unencumbering them from copyright and distribution rights for the OS. Effectively the ISV could create a fully deployable application stack and OS image, then through a dehydration operation the OS image would be removed from the archive leaving just the ISVs application. A customer could then deploy this dehydrated archive and rehydrate the OS from a legally owned copy of the OS repository.

Archives can sometimes be very large in size, dehydration offers a nice solution in requiring archives to take up less space on a system and therefore allows better storage management for multiple archives.

In case customers have some dehydrated archives that they wish to deploy across many systems, a `rehydrate` subcommand would become useful. By rehydrating a dehydration archive back to its normal hydrated state, you can minimize deployment time since a hydrated archive takes less time to deploy than a dehydrated archive.

For more information, see the `archiveadm(8)` man page.

Support for `cloudbase-init`

The `cloudbase-init` service performs initial configuration of guest operating systems in the cloud. These tasks include user creation, password generation, static networking configuration, hostname, SSH public keys, and user data scripts.

The Oracle Solaris 11.4 version of `cloudbase-init` is a Service Management Facility (SMF) service (`application/cloudbase-init`) delivered by the `cloudbase-init` IPS package.

- The `cloudbase-init` package is not installed by default. Install the package only into images that will be deployed in cloud environments.
- The service is enabled by default.
- The configuration file, `/etc/cloudbase-init.conf`, enables only the `UserData` plugin.

Scripts that are exported through user data typically perform system and application configuration tasks that require privileged access. Therefore, the `cloudbase-init` service runs as the user `root`, and any user data scripts must also run as `root`.

Boot Oracle Solaris through Boot Pools over iSCSI-iSER

With this feature, boot pools can use iSCSI-iSER as the default transport protocol instead of iSCSI-IPoIB to boot Oracle Solaris. The server can use boot pools to boot firmware-inaccessible storage devices over iSCSI targets. Using the iSCSI-iSER protocol provides the following benefits:

- Boots Oracle Solaris faster than on iSCSI-IPoIB

- Accesses `rpool` though iSCSI-iSER with:
  - Higher throughput
  - Low transport latency
  - Low CPU utilization
- Requires zero configuration

UEFI Secure Boot

UEFI Secure Boot on Oracle Solaris x86 enables you to install and boot Oracle Solaris on platforms where UEFI Secure Boot is enabled. This feature provides more security by maintaining a chain of trust during boot: digital signatures of the firmware and software are verified before executing the next stage. No break occurs in the chain because of unsigned, corrupt, or rogue firmware or software during the boot process. This feature helps assure that the firmware and software used to boot Oracle Solaris on a hardware platform is correct, and has not been modified or corrupted.

For more information, see *Securing Systems and Attached Devices in Oracle Solaris 11.4*.

Enhancements for Developers

This section describes enhancements for developers that are new in this release that make developing applications on the Oracle Solaris platform easier with state-of-the-art libraries and reliable frameworks.

C11 Programming Language Standard Support

Oracle Solaris 11.4 contains support for the C11 programming language standard: "ISO/IEC 9899:2011 Information technology - Programming languages - C". C". The C11 standard is a compatible revision of the C99 standard. Historically, the C programming language standard has been part of the Single UNIX Specification. However, the new C programming language standard, C11, is available separate from and ahead of the next UNIX specification.

Oracle Solaris 11.4 provides support for C11 alongside C99 to provide customers with C11 support ahead of its inclusion in a future UNIX specification. The new standard can be used with the Developer Studio 12.5 or 12.6, GCC 5, and LLVM/Clang 3.8 C compilers. Developers can also write C programs using the newest available C programming language standard.

Process Control Library

Oracle Solaris 11.4 includes a new process control library, `libproc`, which provides a higher-level interface to features of the `/proc` interface. The library also provides access to information such as symbol tables, which are necessary for the examination and control of processes and threads.

A controlling process using `libproc` can typically:

- Grab a victim process, suspending its execution
- Examine the state of the victim process
- Examine and modify the address space of the victim process
- Make the victim process execute system calls on behalf of the controlling process

- Release the victim process to run again unmolested

The `libproc` library provides all the mechanisms needed by a breakpoint debugger to do its job. It also facilitates the creation of simple one-shot controlling applications to do simple things to victim processes without the processes being aware of the intrusion.

For more information, see the `libproc`(3LIB) man page.

Improved Locale Support

Oracle Solaris 11.4 includes improvements to the existing locale support APIs provided by `libc` and the addition of new APIs defined in the UNIX V7 standard. `uselocale` and other APIs defined in the UNIX V7 standard have been introduced to support locales between threads and changing the locale of the thread. In combination with the new APIs, the existing locale support APIs have been updated to be fully MT-safe. The underlying locale handling in `libc` has been improved for better performance, and smaller resource usage of applications that use multiple locales.

For more information, see the `uselocale`(3C), `newlocale`(3C), `freelocale`(3C), `wctype`(3C), and `localedef`(1) man pages.

User-Mode Watchpoints

Oracle Solaris 11.4 now implements user-mode watchpoints with Silicon Secured Memory (SSM) instead of virtual memory maps. A watchpoint is an event that is triggered when a memory location is written to or read from, and can be used for debugging and performance analysis. Watchpoints are currently implemented by making the page containing the address inaccessible. This action greatly slows execution if the thread frequently touches unrelated locations on the same page. By contrast, SSM has a much finer granularity with the 64-byte cache-line. In addition, SSM is multi-threaded, whereas all threads share the same virtual memory pages.

For more information, see the `dbx(1)` and `mdb`(1) man pages.

DTrace Library

Oracle Solaris 11.4 includes a new process control library, `libdtrace`, that enables developers to write `bespoke` DTrace applications.

For more information, see Appendix A, “libdtrace API Reference,” in *Oracle Solaris 11.4 DTrace (Dynamic Tracing) Guide*.

DWARF Support in DTrace

In Oracle Solaris 11.4, DTrace can use DWARF to perform address to source code metadata translation for user processes. The new `uresolve` option enables the `ustack`, `uaddr`, and `printf` actions of DTrace to translate user addresses to source code file names and line numbers, given the presence of DWARF debugging information. This feature provides a more intuitive interpretation of common diagnostic output while retaining compatibility with common compiler standards.

For more information, see the `ustack`, `uaddr`, and `printf` actions in “Data Recording Actions” in *Oracle Solaris 11.4 DTrace (Dynamic Tracing) Guide*.

`pstack` Support for DWARF-encoded Line Numbers

In Oracle Solaris 11.4, the `pstack` command will annotate frames with source code metadata, given the presence of DWARF debugging information. This feature provides a more intuitive interpretation of common diagnostic output while retaining compatibility with common compiler standards.

For more information, see `pstack` in the `proc(1)` man page.

DWARF Unwinding in `pstack` and `mdb`

In Oracle Solaris 11.4, both `pstack` and `mdb` support DWARF and DWARF-style stack unwinding for user processes. In addition, `pstack` and `mdb` also allow the recovery of function arguments from processes compiled with the new `-preserve_argvalues=complete` option of Oracle Developer Studio. This capability provides particular improvements to the observability and diagnosability of `amd64` processes, though the functionality applies to both 32- and 64-bit processes on both x64 and SPARC.

The `mdb ::stackregs dcmd` is now enabled for `amd64`, where frames' registers are recovered using DWARF-style unwind tables.

Cython

Cython is an optimizing static compiler for both the Python programming language and the extended Cython programming language which is based on Pyrex. This feature enables generation of high performance code using Python.

For more information, see Cython C-Extensions for Python.

Oracle Database Programming Interface-C

Oracle Database Programming Interface-C (ODPI-C) is a wrapper around the Oracle Call Interface (OCI), working transparently with different versions of the Oracle Instant Client libraries.

ODPI-C eliminates the requirement for applications to set `LD_LIBRARY_PATH` prior to execution. Additionally, ODPI-C works transparently with multiple versions of the Oracle Instant Client libraries. Software that requires use of the Oracle Instant Client libraries will not require setting `ORACLE_HOME` prior to execution.

To use ODPI-C, install the `developer/oracle/odpi` package.

For more information about ODPI-C, see the Oracle Database Programming Interface for Drivers and Applications project on GitHub and the `libodpic`(3LIB) man page. Oracle Instant Client is available in IPS format, and no visit to OTN to download zipfiles is necessary.

`cx_Oracle` Python Module

`cx_Oracle` is a Python module that enables you to access Oracle Database 12c and 11i from Python applications. While the module is commonly available through prebuilt packages for other operating systems, it has not until now been similarly available for Oracle Solaris. The module is available in both 32-bit and 64-bit forms. The Oracle Solaris packaged version 5.2 can be used with Python 2.7 and 3.4.

For more information, see cx_Oracle's documentation.

Additional New Features

This section describes additional new features in this release. These features and enhancements add to the existing exhaustive collection of utilities, services, and tools that facilitate enhanced productivity.

GNOME Desktop Environment

The Oracle Solaris desktop environment has been updated from GNOME 2.30 to GNOME 3.24 in Oracle Solaris 11.4. Core desktop applications also have been updated. Some optional applications are no longer included in Oracle Solaris. Users can choose between the modern GNOME Shell environment or the GNOME Classic environment on the login screen. For more details see *Oracle Solaris 11.4 Desktop*.

Oracle Solaris 11.4 no longer supports a multi-level labeled Trusted Extensions desktop and no longer supports Sun Ray configurations. For more details see End of Features (EOF) Planned for Future Releases of Oracle Solaris.

`man` Command Enhancements

In Oracle Solaris 11.4, man pages using the System V sections have been renumbered to the standard sections. The sections `1m`, `4`, `5`, `7`, and their subsections used in the previous releases are now `8`, `5`, `7`, `4`, and their subsections respectively. Users who are familiar with other platforms such as BSD, Linux, or MacOS X can use the same section numbers with the `man` command.

The `man` command supports searching of the hierarchical man page name, which contains one or more slashes. For example, `man system/name-service/switch` can display the man page from `usr/share/man/man8s/system/name-service/switch.8s`.

The `man` command also supports the abbreviation style for the subsections. For example, the `printf(3c)` man page can be searched by either by using `man printf.3` or `man -s 3 printf`.

The `man` command can map the old System V section numbers to the standard ones when necessary, to help find references from older documentation.

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

Oracle Solaris Online Documents for RAD and Web Dashboard

The Oracle Solaris Online Documents for RAD and Web Dashboard application displays searchable documents in the Web Dashboard. To access documentation served by this application, use the Web Dashboard Application menu. Documents are installed by IPS packages to the following location: `/usr/lib/webui/htdocs/solaris/apps/docs`.

The Web Dashboard is the only access point for RAD API documentation. For documents that might be available from multiple locations, Web Dashboard access is very convenient.

For more information, see the `odoc-bundle(5)` and `odoctool(1)` man pages.

`paps` Print Filter

`paps` converts text to PostScript language using the Pango library. `paps` reads an input file and writes a PostScript language or user specified format rendering of the file to standard output. `paps` accepts

international text in any locale and provides internationalized text layout including text shaping and bidirectional text rendering.

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

`iconv` Framework Modernization

The Oracle Solaris `iconv` command and `iconv()` API have been modernized to use the `cconv` API internally for conversion. `cconv` is a unified conversion mechanism that uses Unicode as an intermediate encoding. `cconv` supports a wide range of codesets and provides the `geniconvtbl` command to generate conversion tables with customized conversion rules.

For more information, see the `iconv(1)`, `iconv(3C)`, `cconv(3C)`, `cconv_open(3C)`, `cconv_close(3C)`, `cconvctl(3C)`, `geniconvtbl(1)`, and `geniconvtbl-cconv(5)` man pages.

Locale Name Fallback Mechanism

This new Oracle Solaris 11.4 feature extends the search capability of the `gettext` command to additional directories. When locating a message catalog, the `gettext` command also searches fallback directories based on language and territory, such as fr_FR and fr, in addition to fr_FR.UTF-8. This feature enables the `gettext` command to behave in a way similar to other UNIX-like operating systems.

For more information, see the `gettext(3C)` man page.

Open Group UNIX V7 Product Standard Support

Oracle Solaris 11.4 is certified to conform to The Open Group UNIX V7 Product Standard. This certification is significantly enhanced over the previous UNIX 03 certification. The most significant change is the alignment with the Single UNIX Specification, Version 4, that includes The Open Group Base Specifications, Issue 7, and approved Technical Corrigenda. This version also enables certified systems to support role-based access control as an option.

When using the Oracle Developer Studio 12.5 or 12.6 C compiler, or the latest versions of `gcc`, or LLVM or `clang` found in Oracle Solaris 11.4, Oracle Solaris 11.4 supports:

- The ANSI X3.159-1989 Programming Language - C and ISO/IEC 9899:1990 Programming Language - C (C) interfaces
- ISO/IEC 9899:1990 Amendment 1:1995: C Integrity
- ISO/IEC 9899:1999 Programming Languages - C
- INCITS/ISO/IEC 9899:2011 Programming Languages - C

For more information, see IEEE Std 1003.1TM-2008/The Open Group Technical Standard Base Specifications, Issue 7. You can also see the `standards(7)` man page.

Unicode 8.0 Support

Oracle Solaris UTF-8 locales have been updated to version 8.0 of the Unicode standard. Unicode is a computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems.

CLDR 28 Update

Oracle Solaris 11.4 locales have been updated to CLDR version 28, which provides improvements to locale data quality. For more information, see the *International Language Environments Guide for Oracle Solaris 11.4*.

What's New in Oracle Solaris 11.4

Part No: E60974

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Référence: E60974

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.