Deploying SAS® 9.4 with Oracle Solaris 11.2 Kernel Zones and Unified Archives

by Maureen Chew

Why would you want to run SAS environments in Oracle Solaris 11.2 kernel zones? This article explains the benefits and shows how kernel zones can be easily provisioned and duplicated using Oracle Solaris Unified Archives.

Published May 2014

Table of Contents

About Oracle Solaris Kernel Zones

Oracle Solaris 11.2 kernel zones complement other virtualization technologies such as physical domains (PDOMs on Oracle's SPARC M5-32 and SPARC M6-32 servers), physical partitions (PPARs on Oracle's Fujitsu M10 Servers), Oracle VM Server for SPARC or logical domains (LDOMs), and Oracle Solaris Zones.

Oracle Solaris kernel zones are similar to regular zones in that they run as a "guest" to the host OS. They do not share a running kernel with the host kernel; instead, they run as a type 2 virtual machine manager (VMM) guest to the host (global) Oracle Solaris instance. Type 2 VMM guests run on top of an OS (the host) instead of directly on bare metal.

Oracle Solaris kernel zones provide the capability to have independent kernels and independent upgrade and patch levels. In other words, you can have separate OS versions independent of the global zone. Oracle Solaris 11.2 is a minimum requirement for a kernel zone; however, an Oracle Solaris 10 zone can be created within a kernel zone. Also, it is anticipated that Oracle Solaris 11.2 should be able to host Oracle Solaris 12 kernel zones.

Kernel zones are supported on current sun4v architectures—for example, servers based on Oracle's SPARC T4, SPARC T5, SPARC M5, or SPARC M6 processors and Fujitsu M10 Servers—and x64 architectures—for example, servers based on Intel Nehalem (VT-x with EPT) or AMD Barcelona (AMDv with NPT). Both SAS 9.3 and 9.4 are supported on Oracle Solaris 10 or later for both SPARC- and x64-based architectures.

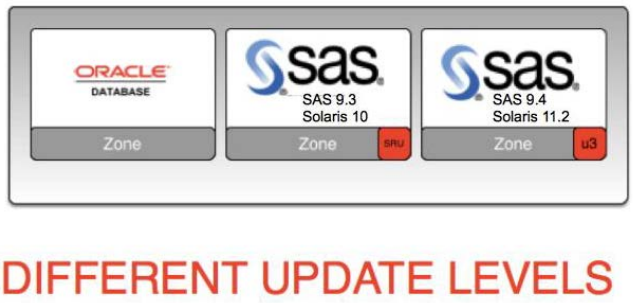

Figure 1. Example architecture in which different versions of SAS and Oracle Solaris are hosted on the same system

Why Are Kernel Zones Relevant for SAS Deployments?

The following are situations in which the features of kernel zones are especially appropriate or even required for SAS application deployments:

- When you can't consolidate other applications that have different OS requirements
- When you need direct and exclusive access to storage devices
- When you need access to Oracle Solaris DTrace or other kernel-based performance and profiling tools
- When you need access to kernel features, such as the ability to change the date or time settings, or you need access to `/dev` structures
- When you want to avoid having to learn new interfaces, because the same CLI commands used for zones work for kernel zones

Kernel zones in combination with Oracle Solaris 11 boot environments (BEs) enable a single server to function as a consolidated test server in production/test/development environments, allowing for a simultaneous, yet independent, patching and upgrading process for multiple environments on a single system.

Additionally, subsequent deployment to production systems for complicated upgrades or test scenarios can be easily facilitated (for example, a proven test environment can be cloned and redeployed as a full kernel zone unit). Also, if a problem occurs on a production system, the environment can be replicated in a much easier fashion by cloning and provisioning the environment to a test system for debugging purposes.

Minimal Performance Impact

As shown in Figure 2, performance tests for SAS testing run in a kernel zone showed little to no overhead on baseline tests. A series of 40 standard SAS tests that are CPU-, I/O- and memory-intensive were run first on the global zone and then in the kernel zone. The SAN LUNs, which were configured in a ZFS pool, were initially attached to the global zone but then exported, attached, and reimported into the kernel zone so that the I/O configuration was identical in each run.

The SAS programs were all set to `CPUCOUNT=4` and `memsize=512M` (with `SASWORK` going to the same SAN LUN).

Although no real performance impact was noted between the two runs in either the total overall time or within the individual tests, your results might differ.

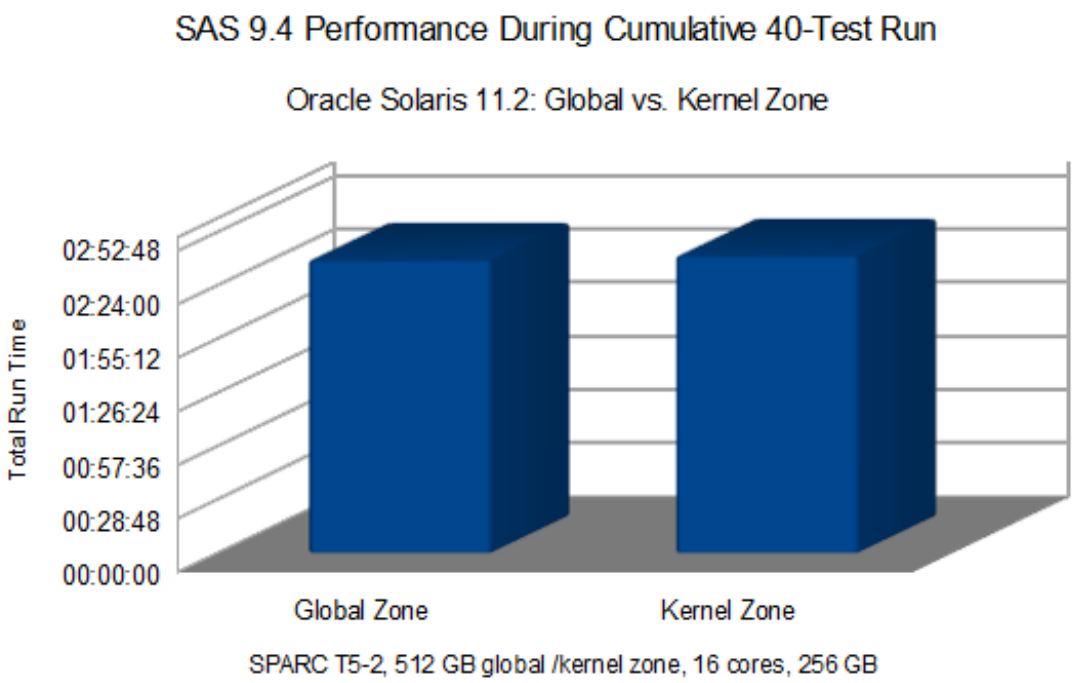

Figure 2. Results of performance tests

Steps for Creating a Kernel Zone

Overview

It's very easy to create a kernel zone. Kernel zones allow for dedicated private storage with direct installation of kernel device drivers.

The example below shows how simple it is to create a kernel zone. The following are the basic steps:

- Unmount and export a ZFS pool that you wish to use (this is an optional step and is not shown).
- Create the new kernel zone and provision it.
- Perform post-installation tasks, such as booting the zone; configuring the network identity, root password, default user, name services, directory services, time zone, and date/time settings for the new kernel zone; and importing the ZFS pool.

Once the new kernel zone is up and running, the power of the Unified Archives feature or Oracle Solaris 11.2 can be used to easily replicate the environment.

The following hardware environment was used for the example in this article:

- Oracle's SPARC T5-2 server, 2 x 3.6 GHz (per socket:16 core, 8 threads/core, 128 virtual CPUs)
- 512 GB RAM
- Oracle Solaris ZFS pools configured on 3PAR SAN storage

The global zone, `t5-ptest1`, was upgraded from Oracle Solaris 11.1 to Oracle Solaris 11.2 using the easy and quick steps shown in Appendix A. Oracle Solaris 11 introduced boot environments (BEs), which are facilitated by using Oracle Solaris ZFS for boot disks. Using Oracle Solaris 11 BEs and performing subsequent upgrades do not require upfront planning or complicated procedures.

For the upgrade, a new BE was created and the Oracle Solaris 11.2 upgrade was performed on this new BE. Once completed, the new BE was marked to become active on the next reboot. A simple and quick reboot was the only downtime needed for the upgrade. If the upgrade hadn't worked, only one simple command would have been needed to reactivate the previous, original BE. BEs are powerful and very appreciated by systems administrators.

Prerequisites

Before attempting to create a kernel zone, use the `virtinfo`(1M) command to verify that the appropriate hardware, firmware, and OS levels are in place and that `kernel-zone` is listed in the output.. For servers based on SPARC T4 and SPARC T5 processors, a newer firmware revision might be required.

```
$ virtinfo
NAME            CLASS
logical-domain  current
non-global-zone supported
kernel-zone     supported
logical-domain  supported
```

Creating and Provisioning the Kernel Zone

Use the commands shown in Listing 1 to create a new kernel zone named `t5kz1`. In the script, the publisher is set to a local repository mounted on `/repo`. When the `zoneadm install` command is issued, the kernel zone is provisioned from the global zone's default publisher.

Note: The default configuration for a kernel zone is one (virtual) CPU, 2 GB RAM, a 16 GB boot disk, and a single NIC with a random MAC address. These defaults are increased in our example.

```
#!/bin/sh
ZN=t5kz1
set -x

echo Configuring zone $ZN
zonecfg -z $ZN -f zone3.txt
echo
echo Installing zone $ZN
zoneadm -z $ZN install -x install-size=20g

echo
zoneadm -z $ZN boot
```

Listing 1. Commands for creating the kernel zone

By modifying the commands in Listing 1, you can set the system identity/profile automatically. (See the example `/usr/share/auto_install/sc_profiles/sc_sample.xml` file on the system on which Oracle Solaris is installed.)

To do this, instead of using this command:

```
zoneadm -z $ZN install -x install-size=20g
```

use the following command, which contains the `–c` specification:

```
zoneadm -z $ZN install -x install-size=20g –c
/usr/share/auto_install/sc_profiles/sc_mysite.xml
```

The code in Listing 1 refers to a zone configuration file called `zone3.txt`, which is shown in Listing 2. It performs the following customizations:

- Sets memory allocation to 256 GB
- Sets the virtual CPUs to 128 or 16 cores (1 socket for servers based on the SPARC T5 processor)
- Brings in the recently exported ZFS pool LUNs

```
create -b -t SYSsolaris-kz
set autoboot=true
select capped-memory
set physical=256G
end

add dedicated-cpu
set ncpus=128
end

add device
set match=/dev/rdsk/c14t21220002AC001593d8
end
add device
set match=/dev/rdsk/c14t21220002AC001593d9
end
add device
set match=/dev/rdsk/c14t21220002AC001593d10
end
add device
set match=/dev/rdsk/c14t21220002AC001593d11
end
add device
set match=/dev/rdsk/c14t21220002AC001593d12
end
add device
set match=/dev/rdsk/c14t21220002AC001593d13
end
add device
set match=/dev/rdsk/c14t21220002AC001593d14
end
add device
set match=/dev/rdsk/c14t21220002AC001593d15
end

verify
commit
exit
```

Listing 2. `zone3.txt` file

The results from the creation of the kernel zone are shown in Figure 3:

```
File  Edit  View  Terminal  Help
+ echo Configuring zone
Configuring zone
+ zonecfg -z t5kz1 -f zone3.txt
+ echo

+ echo Installing zone t5kz1
Installing zone t5kz1
+ zoneadm -z t5kz1 install -x 'install-size=20g'
Progress being logged to /var/log/zones/zoneadm.20140422T032436Z.t5kz1.install
pkg cache: Using /var/pkg/publisher.
 Install Log: /system/volatile/install.28127/install_log
 AI Manifest: /tmp/zoneadm27569.ZZaKmB/devel-ai-manifest.xml
  SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
Installation: Starting ...

        Creating IPS image
        Installing packages from:
            solaris
                origin:  file:///repo/repo/
        The following licenses have been accepted and not displayed.
        Please review the licenses for the following packages post-install:
          consolidation/osnet/osnet-incorporation
        Package licenses may be viewed using the command:
          pkg info --license <pkg_fmri>

DOWNLOAD                                PKGS         FILES    XFER (MB)   SPEED
Completed                           544/544   76792/76792  677.7/677.7    0B/s

PHASE                                          ITEMS
Installing new actions                 103987/103987
Updating package state database                 Done
Updating package cache                           0/0
Updating image state                            Done
Creating fast lookup database                   Done
Installation: Succeeded
        Done: Installation completed in 233.471 seconds.


+ echo

+ zoneadm -z t5kz1 boot
```

Figure 3. Results from creating the kernel zone

Post-Installation Tasks

Once the code in Listing 1 is done and it boots the zone, run the `zlogin -C t5kz1` command to enable a console login. At this point, the system will run `sysconfig`(1M), which enables you to configure the network identity, root password, default user, name services, directory services, time zone, date/time settings, and so on.

The previously existing ZFS pool that was used in example configuration for this article can be imported using the following command, and the ZFS file system will be automatically mounted on the same, previously used mount point.

```
root# zpool import
```

In order to run the SAS Display Manager System (DMS), the Motif package, which hosts `libXm.so`, must be separately installed. By default, this package is not installed even if it is installed on the global zone at the time the kernel zone is created. To install this package, run the following command:

```
root# pkg install motif
```

Another post-installation task is to consider how much memory should be allocated for the ZFS file system cache (the Adaptive Replacement Cache [ARC]). There is no general rule of thumb; the defaults are reasonable, but you should holistically consider the management of large memory consumers. To set this parameter, set `zfs_arc_max` to half of what you want ZFS to use, for example:

```
# echo "set zfs:zfs_arc_max=0x40000000" >> /etc/system
# reboot
```

Cloning and Deployment Using Oracle Solaris Unified Archives

To create a clone, the following example was run from the global zone (`t5-ptest1`). The resulting archive was deployed on the same SPARC T5-2 server for demonstration purposes, but it can be deployed on any other system with a similar architecture (SPARC, in this case) that supports kernel zones.

The Unified Archive carries all provisioning information with it in the form of a ZFS send-receive payload stream. This is a very powerful feature and does not require the originating system/zone and the receiving system/zone to have the same OS revision level. Figure 4 shows the flexibility that the Unified Archive feature provides in terms of deployment options.

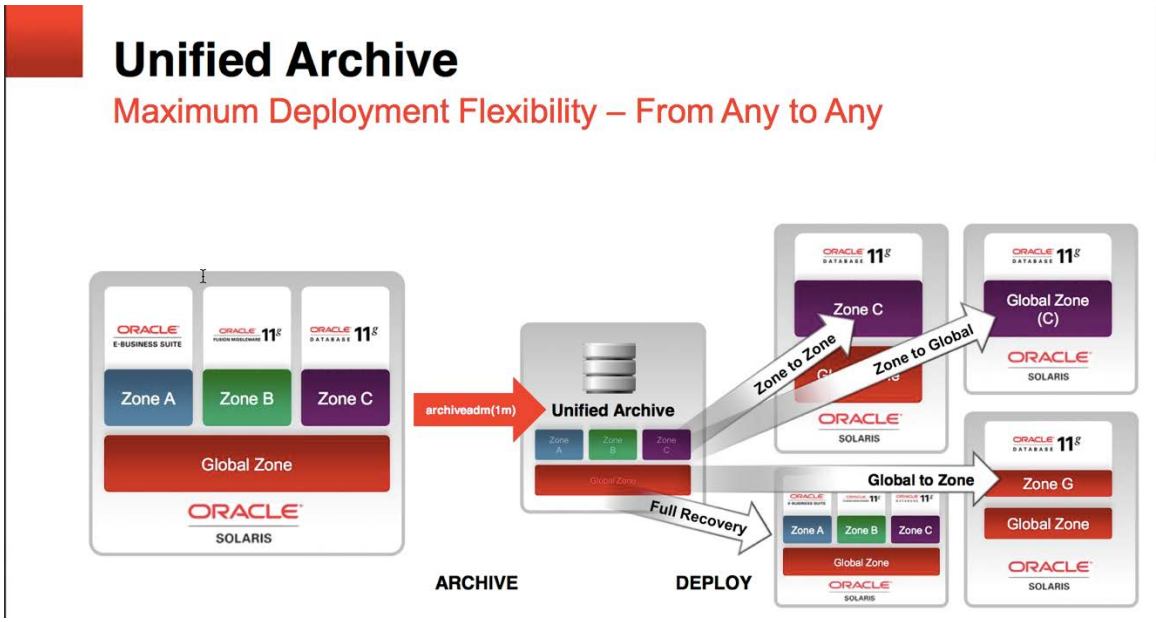

Figure 4. Deployment options provided by Unified Archives

Use the `archiveadm`(1M) command to manage Unified Archives. There are two types of archives:

- A *clone archive,* which contains only the latest BE and is in the "sysunconfig" state
- A *recovery archive,* which contains all BEs and retains the "sysconfig" state

When creating an archive of the kernel zone (`t5kz1`), the zone itself must be up and running. In the following example, we're creating a clone archive, not a recovery archive.

Creating a Clone

To create a clone, run the commands shown in Listing 3:

```
root# archiveadm create /z0/t5kzclone.uar -z t5kz1

Initializing Unified Archive creation resources...
Unified Archive initialized: /z0/t5kzclone.uar
Logging to: /system/volatile/archive_log.6260
Executing dataset discovery...
Dataset discovery complete
Creating install media for zone(s)...
Media creation complete
Preparing archive system image...
Beginning archive stream creation...
Archive stream creation complete
Beginning final archive assembly...
```

```
Archive creation complete


root# ls –l /z0/t5kzclone.uar
total 12074523
-rw-r--r--   1 root     root     6177536000 Apr 24 15:47
t5kzclone.uar


root# archiveadm info t5kzclone.uar
Archive Information
          Creation Time:  2014-04-24T19:30:35Z
            Source Host:  t5-ptest1
           Architecture:  sparc
       Operating System:  Oracle Solaris 11.2 SPARC
      Deployable Systems: t5kz1

root# archiveadm info -v t5kzclone.uar
Archive Information
          Creation Time:  2014-04-24T19:30:35Z
            Source Host:  t5-ptest1
           Architecture:  sparc
       Operating System:  Oracle Solaris 11.2 SPARC
       Recovery Archive:  No
              Unique ID:  265382ed-a859-4a29-aaac-
cc4815632a1c
        Archive Version:  1.0

Deployable Systems
           't5kz1'
             OS Version:  0.5.11
              OS Branch:  0.175.2.0.0.36.1
              Active BE:  solaris-3
                  Brand:  solaris-kz
            Size Needed:  12.8GB
              Unique ID:  7cdf3f2f-bae6-cc99-a565-
dc27a2f03139
               AI Media:  0.175.2_ai_sparc.iso
```

Listing 3. Creating a clone

At this point, you should shut down the original kernel zone, `t5kz1`, as you provision the second kernel zone (`t5kz2`), since `tkz1` is provisioned to take up approximately half the system's resources. `t5z2` is configured to be the same size as `tkz1` and resources would be overcommitted if both were up at the same time.

Run the following command to shut down `t5kz1` from the global zone:

```
root# zoneadm –z t5kz1 shutdown
```

Deploying the New Kernel Zone

Only two commands are required to deploy the zone clone from the Unified Archive, `zonecfg` and `zoneadm`, as shown in Listing 4. Note the two `-z` parameters in the following `zonecfg` and `zoneadm` commands: `t5kz2` is the new zone and `t5kz1` is the original zone. Unified Archives can archive all the zones or any subset of them. Although our archive contains only one zone, the second `–z` specifies the zone of interest that is being cloned.

```
#!/bin/sh
set -x

zonecfg -z t5kz2 create -a /z0/t5kzclone.uar -z t5kz1
echo
zoneadm list -icv
echo
zoneadm -z t5kz2 install -x install-size=60g -a
/z0/t5kzclone.uar -z t5kz1
```

Listing 4. Deploying the clone

Additionally, note the root `install-size` parameter is larger than in the original deployment. Some things that are not addressed in the simplified examples shown here are the properly sized areas and locations for the swap and dump devices. For a 256 GB zone, the initial swap configuration of 5 GB is unlikely to be sufficient for SAS. When SAS does memory allocation via `mmap`(2), a swap reservation needs to be made regardless of whether any swapping will take place. Thus, a swap size that is in proportion to the RAM configuration needs to be taken into consideration. Similarly, the dump devices are generally allocated on file systems that are not co-located with the root partition. Deployment through Unified Archives attempts to accommodate the dump device based on the memory configuration on the root file system.

Summary

For SAS environments, Oracle Solaris kernel zones and Unified Archives can provide great flexibility for resolving common enterprise IT problems, such as the following:

- The need to consolidate applications in environments where different—and often conflicting—revision and patch levels are required
- The need to easily transfer test environments to production (and vice versa)

These capabilities create a robust and full complement of virtualization options that are easy to deploy.

The performance impact of running SAS within a kernel zone appears to be minimal. The performance achieved when running SAS within a kernel zone is comparable to the performance achieved when running SAS in a traditional Oracle Solaris Zone.

Appendix A: Upgrading from Oracle Solaris 11.1 to Oracle Solaris 11.2

Run the commands shown in Listing 5 to create an extra backup BE and then perform the upgrade:

```
root# beadm create solaris11.1
root# beadm list
BE                 Active Mountpoint Space  Policy Created
--                 ------ ---------- -----  ------ -------
solaris            -      -          8.49M  static 2013-06-27 17:33
solaris-1          NR     /          43.73G static 2013-10-17 01:29
solaris-1-backup-1 -      -          64.0K  static 2013-10-18 23:25
solaris11.1        -      -          197.0K static 2014-04-10 13:02

root# pkg publisher  set-publisher  -g file:///repo/repo solaris

root# pkg publisher
PUBLISHER                   TYPE     STATUS P LOCATION
solaris                     origin   online F file:///repo/repo/

root# pkg update --accept  entire
Refreshing catalog 1/1 solaris...
Creating Plan (Solver setup)...
Creating Plan (Finding local manifests): ...
Creating Plan (Download Manifests   0/807)...
Creating Plan (Committing Manifests): ...
Creating Plan (Package planning:   1/811): ...
Creating Plan (Merging actions): ...
Creating Plan (Checking for conflicting actions): ...
Creating Plan (Consolidating action changes): ...
Creating Plan (Evaluating mediators): ...
            Packages to remove:   4
           Packages to install:  65
            Packages to update: 742
           Mediators to change:   1
       Create boot environment: Yes
Create backup boot environment:  No
...

DOWNLOAD                                  PKGS         FILES    XFER (MB)   SPEED

data/docbook                            0/811       0/38051    0.0/803.2      --
data/docbook                            1/811       0/38051    0.0/803.2      --
system/scheduler/fss                    1/811       0/38051    0.0/803.2      --
system/scheduler/fss                    2/811       0/38051    0.0/803.2      --
file/slocate                            2/811       0/38051    0.0/803.2      --
file/slocate                            3/811       0/38051    0.0/803.2      ---
...
x11/xfs                               806/811   38013/38051  803.1/803.2   cache
x11/xfs/xfs-utilities                 806/811   38013/38051  803.1/803.2   cache
x11/xkill                             807/811   38026/38051  803.1/803.2   cache
x11/xlock                             808/811   38032/38051  803.1/803.2   cache
x11/xmag                              809/811   38037/38051  803.2/803.2   cache
x11/xvidtune                          811/811   38051/38051  803.2/803.2   cache
Completed                             811/811   38051/38051  803.2/803.2    0B/s
```

```
PHASE                                          ITEMS

Removing old actions                             1/10476
Removing old actions                           942/10476
...
Installing new actions                        7727/23704
...
Updating modified actions                        1/36399
...
Updating package state database                  working
...
Updating package state database                     Done
Updating package cache                           1/746
...
Updating image state                                Done
...
Creating fast lookup database                       Done

A clone of solaris-1 exists and has been updated and activated.
On the next boot the Boot Environment solaris-2 will be
mounted on '/'.  Reboot when ready to switch to this updated BE.

root# beadm list

BE                 Active Mountpoint Space  Policy Created
--                 ------ ---------- -----  ------ -------
solaris            -      -          8.49M  static 2013-06-27 17:33
solaris-1          N      /          14.69M static 2013-10-17 01:29
solaris-1-backup-1 -      -          64.0K  static 2013-10-18 23:25
solaris-2          R      -          46.91G static 2014-04-10 13:15
solaris11.1        -      -          197.0K static 2014-04
```

Listing 5. Creating a BE and upgrading

Appendix B: Deleting a Zone

If a mistake is made when provisioning a zone, use the commands shown in Listing 6 to remove the zone:

```
#!/bin/sh
ZN=t5kz2
zoneadm list -icv
set -x
zoneadm -z $ZN uninstall -F
zonecfg -z $ZN delete -F
zoneadm list –icv
```

Listing 6. Removing a zone

See Also

See the Oracle and SAS partner web page.

Also see these additional Oracle Solaris resources:

- Download Oracle Solaris 11.2
- Access Oracle Solaris 11.2 product documentation
- Access Oracle Solaris 11.2 resources
- Access all Oracle Solaris 11 how-to articles
- Learn more with Oracle Solaris 11 training and support
- See the official Oracle Solaris blog
- Check out The Observatory and OTN Garage blogs for Oracle Solaris tips and tricks
- Follow Oracle Solaris on Facebook and Twitter

About the Author

Maureen is a principal software engineer with Oracle sitting onsite at SAS and thinks about all kinds of technology and product convergences between SAS and Oracle products. She is a 26-year veteran with Sun and Oracle and currently resides in Chapel Hill, NC.

Revision 1.0, 05/07/14