Oracle® Solaris Cluster Data Service for Oracle Solaris Zones Guide

Part No: E56738
September 2015

ORACLE

Oracle Solaris Cluster Data Service for Oracle Solaris Zones Guide

Part No: E56738

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: E56738

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.

Contents

Using This Documentation

- **Overview** – Explains how to install and configure Oracle Solaris Cluster HA for Oracle Solaris Zones (HA for Solaris Zones) for `solaris`, `solaris10`, and `solaris-kz` branded zones.
- **Audience** – Experienced system administrators with extensive knowledge of Oracle software and hardware.
- **Required knowledge** – Knowledge of the Oracle Solaris operating system, of Oracle Solaris Cluster software, and expertise with the volume manager software that is used with Oracle Solaris Cluster software.

This document is not to be used as a planning or presales guide.

Product Documentation Library

Documentation and resources for this product and related products are available at `http://www.oracle.com/pls/topic/lookup?ctx=E56676-01`.

Feedback

Provide feedback about this documentation at `http://www.oracle.com/goto/docfeedback`.

♦ ♦ ♦ CHAPTER 1

Installing and Configuring HA for Solaris Zones

This chapter explains how to install and configure Oracle Solaris Cluster HA for Oracle Solaris Zones (HA for Solaris Zones) for Oracle Solaris Zones (`solaris`), Oracle Solaris 10 Zones (`solaris10`), and Oracle Solaris Kernel Zones (`solaris-kz`).

Install and configure this data service to run in the global zone. For updated information about supported configurations of this data service, see the Oracle Solaris Cluster 4 Compatibility Guide (`http://www.oracle.com/technetwork/server-storage/solaris-cluster/overview/solariscluster4-compatibilityguide-1429037.pdf`).

This chapter contains the following sections.

HA for Solaris Zones Overview

The `solaris` and `solaris10` brands of non-global zones as well as the `solaris-kz` branded zones are supported for configuration with the HA for Solaris Zones data service.

A non-global zone is a complete runtime environment for applications that run on the Oracle Solaris Operating System. Oracle Solaris Resource Manager and Solaris Zones software

partitioning technology are both parts of Oracle Solaris Zones. These components address different qualities the zone can deliver and work together to create a complete zone. The zones portion provides a virtual mapping from the application to the platform resources. Non-global zones allow application components to be isolated from one application even though the zones share a single instance of the Oracle Solaris Operating System. Resource management features permit you to allocate the quantity of resources that a workload receives.

The Solaris Zones facility in the Oracle Solaris Operating System provides an isolated and secure environment in which to run applications on your system. When you create a zone, you produce an application execution environment in which processes are isolated from the rest of the system.

This isolation prevents processes that are running in one zone from monitoring or affecting processes that are running in other zones. Even a process that is running with superuser credentials cannot view or affect activity in other zones. A zone also provides an abstract layer that separates applications from the physical attributes of the machine on which they are deployed. Examples of these attributes include physical device paths.

Every Oracle Solaris system contains a host global zone. The host global zone is both the default zone for the system and the zone that is used for system-wide administrative control. The `solaris` and the `solaris10` brands of non-global zones and the `solaris-kz` branded zones are created by the administrator of the host global zone.

For more information about the `solaris`, `solaris10`, and `solaris-kz` brands, see the following documentation:

- *Introduction to Oracle Solaris Zones*
- *Creating and Using Oracle Solaris Zones*
- *Creating and Using Oracle Solaris 10 Zones*
- *Creating and Using Oracle Solaris Kernel Zones*

HA for Solaris Zones enables Oracle Solaris Cluster to manage Solaris Zones by providing components to perform the following operations:

- The orderly booting, shutdown and fault monitoring of a zone through the `sczbt` component.
- The orderly startup, shutdown and fault monitoring of an application within the zone, using scripts or commands through the `sczsh` component.
- The orderly startup, shutdown and fault monitoring of an Oracle Solaris Service Management Facility (SMF) service within the zone through the `sczsmf` component.

You can configure HA for Solaris Zones as a failover service or a multiple-masters service. You *cannot* configure HA for Solaris Zones as a scalable service.

When a Solaris Zone is managed by the HA for Solaris Zones data service, the Solaris Zone becomes a Solaris HA zone or a multiple-masters Solaris Zone across the Oracle Solaris Cluster nodes. The failover in case of a Solaris HA zone is managed by the HA for Solaris Zones data service, which runs only within the global zone.

For conceptual information about failover data services, multiple-masters data services, and scalable data services, see *Oracle Solaris Cluster 4.3 Concepts Guide* .

Note - The use of extension properties eliminate the need for a parameter file for configuring HA for Solaris Zones. For information about the extension properties, see Appendix B, “HA for Solaris Zones Extension Properties”.

Overview of Installing and Configuring HA for Solaris Zones

The following table summarizes the tasks for installing and configuring HA for Solaris Zones and provides cross-references to detailed instructions for performing these tasks. Perform the tasks in the order that they are listed in the table.

TABLE 1 Tasks for Installing and Configuring HA for Solaris Zones

Task	Instructions
Plan the installation	“Planning the HA for Solaris Zones Installation and Configuration” on page 12
Install and configure the Solaris Zones	“Installing and Configuring Zones” on page 17
Install HA for Solaris Zones Packages	“Installing the HA for Solaris Zones Package” on page 34
Register and configure HA for Solaris Zones components	“Registering and Configuring HA for Solaris Zones” on page 35
Verify HA for Solaris Zones installation and configuration	“Verifying the HA for Solaris Zones and Configuration” on page 57
Upgrading the non-global zones managed by HA for Solaris Zones	“Upgrading Non-Global Zones Managed by HA for Oracle Solaris Zones” on page 58
Tune the HA for Solaris Zones fault monitors	“Tuning the HA for Solaris Zones Fault Monitors” on page 58
Tune the HA for Solaris Zones `Stop_timeout property`	“Tuning the HA for Solaris Zones `Stop_timeout` property” on page 60
Debug HA for Solaris Zones	“How to Activate Debugging for HA for Solaris Zones” on page 62

Planning the HA for Solaris Zones Installation and Configuration

This section contains the information you need to plan your HA for Solaris Zones installation and configuration. The configuration restrictions and requirements in the subsections that follow apply only to HA for Solaris Zones.

Caution - Your data service configuration might not be supported if you do not observe these restrictions.

Requirements and Restrictions for Zone Network Addresses

- The configuration of a zone's network addresses depends on the level of high availability (HA) you require for it and the configured `ip-type` option. You can choose between no HA, HA through the use of only public network management (PNM) objects, or HA through the use of PNM objects and `SUNW.LogicalHostName` (`ip-type=shared only`). PNM objects include Internet Protocol network multipathing (IPMP) groups, trunk and datalink multipathing (DLMP) link aggregations, and VNICs that are directly backed by link aggregations.
- Your choice of a zone's network addresses configuration affects some configuration parameters for the zone boot resource. For more information, see "Registering and Configuring HA for Solaris Zones" on page 35.
- The following restrictions apply if `ip-type` is set to `shared`:
  - If HA for the zone's addresses is not required, configure the zone's addresses by using the `zonecfg` utility.
  - If only HA through PNM protection in the global zone is required, configure the zone's addresses by using the `zonecfg` utility and place the zone's addresses on an adapter within a PNM object.

- If HA through PNM protection in the global zone and protection against the failure of all physical interfaces by triggering a failover is required, choose one option from the following list:
  - If you require the `SUNW.LogicalHostName` resource type to manage one or a subset of the zone's addresses, configure a `SUNW.LogicalHostName` resource for those zone's addresses and not by using the `zonecfg` utility. Use the `zonecfg` utility only to configure the zone's addresses that are not required to be under the control of the `SUNW.LogicalHostName` resource.
  - If you require the `SUNW.LogicalHostName` resource type to manage all the zone's addresses, configure a `SUNW.LogicalHostName` resource with a list of the zone's addresses and do not configure them by using the `zonecfg` utility.
  - Otherwise, configure the zone's addresses by using the `zonecfg` utility and configure a separate redundant IP address in the same subnet for use by a `SUNW.LogicalHostName` resource, which must not be configured using the `zonecfg` utility.
- The following restrictions apply if `ip-type` is set to `exclusive`:
  - The SC_NETWORK variable in the `sczbt_config` file must be set to `false` to successfully register the `sczbt` resource.
  - Do not configure a resource dependency on a `SUNW.LogicalHostname` resource from the `sczbt` resource.
  - A `linkname` is required for `anet` resources within `zonecfg`. Set the `linkname` value explicitly instead of using the `auto` option.
- The zone network addresses that are managed by a `SUNW.LogicalHostname` resource get configured for the zone and unconfigured from the zone asynchronously during the boot and shutdown of the zone. An application that uses these network addresses has to be managed by either the `sczsh` component or the `sczsmf` component, to ensure correct order of start and stop of the application with the corresponding network addresses.

  If the application is started by runlevel or SMF services within the zone, without using the `sczsh` or `sczsmf` component, then the network addresses used by that application must be configured using the `zonecfg` utility and must not be managed by a `SUNW.LogicalHostname` resource.

Requirements and Restrictions for an HA Zone

- If the `rootzpool` zone property is not set, the zone path of a non-global zone in an HA Zone configuration must reside on a highly available local file system. The zone must be configured on each cluster node where the zone can reside.

- The zone is active on only one node at a time, and the zone's address is plumbed on only one node at a time. Application clients can then reach the zone through the zone's address, wherever that zone resides within the cluster.
- Ensure that the zone's `autoboot` property is set to `false`. Setting a zone's `autoboot` property to `false` prevents the zone from being booted when the host global zone is booted. The HA for Solaris Zones data service can manage a zone only if the zone is booted under the control of the data service.
- Ensure that the zone configuration defines a generic attribute with name `osc-ha-zone` of type `boolean` and value `true`. This attribute is used by the `svc:/system/cluster/osc-ha-zone-state-cleanup` SMF service on each node to identify a zone controlled by the `sczbt` component. The `svc:/system/cluster/osc-ha-zone-state-cleanup` SMF service must be enabled.
- For a `solaris` branded zone, failover behavior differs depending on the version of Oracle Solaris.
  - On Oracle Solaris 11.2, the last zone boot environment that was booted is first cloned and then activated on the node.
  - On Oracle Solaris 11.3, the zone is attached using the `-x deny-zbe-clone` option of the `zoneadm attach` command. For more information about this option, see the `zoneadm`(1M) man page.
- For a `solaris-kz` branded zone, observe the following restrictions:
  - You cannot specify the `Mounts` variable within the `sczbt` configuration file.
  - You cannot set the `SC_NETWORK` variable to `true` within the `sczbt` configuration file.
- For a `solaris-kz` branded zone set with `Migrationtype=live`, a live migration of a kernel zone is performed over the cluster private interconnect. The migration uses the `ssh` protocol that is specified in the RAD URI using the default RAD port. A passwordless `ssh` login for the `root` user is used between the cluster nodes over the cluster interconnect.

  To support this behavior, the following SMF services must be enabled on all cluster nodes:
  - `svc:/system/rad:local`
  - `svc:/system/rad:remote`
  - `svc:/network/kz-migr:stream`
- In some cases where the cluster cannot determine the target node to which the HA for Solaris Zones resource group is live migrating, it uses an ordinary resource group switchover instead of using live migration. In such cases, the kernel zone shuts down on its current node and then boots on its new node.

  To achieve live migration in such cases, relocate the HA for Solaris Zones resource group by using the `clresourcegroup switch` command explicitly on the resource group, rather than depending on node evacuation or strong resource group affinities to move the resource group.

- For a `solaris-kz` branded zone that is set with either `Migrationtype=warm` or `Migrationtype=live`, to successfully migrate a kernel zone between different CPU types, you must set the `cpu-arch` zone property. For more information about the `cpu-arch` property, see .“solaris-kz SPARC Only: Cross-CPU Migration” in *Oracle Solaris Zones Configuration Resources* .

Requirements and Restrictions for a Multiple-Masters Zone

- The zone path of a zone in a multiple-masters configuration must reside on the local disks of each node. The zone must be configured with the same name on each node that can master the zone.
- Each zone that is configured to run within a multiple-masters configuration must also have a zone-specific address. Load balancing for applications in these configurations is typically provided by an external load balancer. You must configure this load balancer for the address of each zone. Application clients can then reach the zone through the load balancer's address.
- Ensure that the zone's `autoboot` property is set to `false`. Setting a zone's `autoboot` property to `false` prevents the zone from being booted when the global zone is booted. The HA for Solaris Zones data service can manage a zone only if the zone is booted under the control of the data service.

Requirements and Restrictions for the Zone Path of a Zone

- The zone path of a zone that HA for Solaris Zones manages cannot reside on a global file system.
- If the non-global zone is in a failover configuration, the zone path must either reside on a highly available local file system or the `rootzpool` zone property must be set to point to shared-storage devices. If the storage URI points to a logical unit or iSCSI device, you can use the `SUNW.HAStoragePlus` resource to monitor the corresponding DID device."
- For an Oracle Solaris Kernel Zone, the boot storage is specified as described in the `suri(5)` man page. If the storage URI points to a `zvol`, then the corresponding `zpool` must be managed by a `SUNW.HAStoragePlus` resource. If the storage URI points to a logical unit or iSCSI device, then the `SUNW.HAStoragePlus` resource can be used to monitor the corresponding `did` device.
- If the zone is in a multiple-masters configuration, the zone path must reside on the local disks of each node.

Dependencies Between HA for Solaris Zones Components

The dependencies between the HA for Solaris Zones components are described in the following table:

TABLE 2 Dependencies Between HA for Solaris Zones Components

Component	Dependency
Zone boot resource (`sczbt`)	`SUNW.HAStoragePlus` - In a failover configuration for a non-global zone, if the `rootzpool` zone property is not set, the zone's zone path must be on a highly available file system managed by a `SUNW.HAStoragePlus` resource. If either the `rootzpool` or `zpool` zone property is set and if the storage URI points to a logical unit or to an iSCSI device, you can use the `SUNW.HAStoragePlus` resource to monitor the storage devices that are configured for those zone properties. `SUNW.HAStoragePlus` - In a failover configuration for a kernel zone, if the storage URI points to a logical unit or to an iSCSI device, the `SUNW.HAStoragePlus` resource can be used to monitor the storage devices configured as a boot device or as a suspend device. If the boot device points to a `zvol`, then the corresponding `zpool` is managed by `SUNW.HAStoragePlus`. Similarly, if the suspend device is specified to point to a path, then the storage resource managing the corresponding highly available file system is specified as the resource dependency. `SUNW.LogicalHostName` - This dependency is required only if the zone's address is managed by a `SUNW.LogicalHostName` resource and the `ip-type` is set to shared
Zone script resource (`sczsh`)	Zone boot resource
Zone SMF resource (`sczsmf`)	Zone boot resource

These dependencies are set when you register and configure HA for Solaris Zones. For more information, see “Registering and Configuring HA for Solaris Zones” on page 35.

The `sczbt_register` script defines a resource dependency of type `Resource_dependencies_offline_restart` as follows:

- If you set the `SC_LH` variable within the `sczbt_config` file, then the `Resource_dependencies_offline_restart` property of the `sczbt` component will contain the `SUNW.LogicalHostname` resource name as set with the `SC_LH` variable.
- If you set the `HAS_RS` variable within the `sczbt_config` file, then the `Resource_dependencies_offline_restart` property of the `sczbt` component will contain the storage resource name as set with the `HAS_RS` variable.

When you configure a `solaris-kz` branded zone for warm migration, where the suspend image is hosted on a file system managed by HAStoragePlus or on any other cluster resource

managing that file system, you need to set the `HAS_RS` variable to the corresponding resource name. This ensures that the resource dependency to the storage resource is set up when the `sczbt` resource is registered.

The zone script resource and SMF resource are optional. If used, multiple instances of the zone script resource and SMF resource can be deployed within the same resource group as the zone boot resource. Furthermore, if more elaborate dependencies are required, refer to the `r_properties(5)` and `rg_properties(5)` man pages for further dependencies and affinities settings.

For a kernel zone, if the `sczbt` component is configured with `Migrationtype=warm` or `Migrationtype=live`, it will still perform the start and stop operations on the corresponding services that are managed by the `sczsh` or the `sczsmf` component. If you need to have all the services running within the kernel zone during warm or live migration, do not configure the `sczsh` or the `sczsmf` component for those services.

Installing and Configuring Zones

Installing and configuring Solaris Zones involves the following tasks:

1. Enabling a zone to run in your chosen data service configuration, as explained in the following sections:
   - “How to Enable a Zone to Run in a Failover Configuration” on page 18
   - “How to Enable a Zone to Run in a Multiple-Masters Configuration” on page 20
2. Installing and configuring a zone, as explained in the following sections:
   - “How to Install a `solaris` Branded Zone and Perform the Initial Internal Zone Configuration” on page 20
   - “How to Install a `solaris10` Branded Zone and Perform the Initial Internal Zone Configuration” on page 25
   - “How to Install a `solaris-kz` Branded Zone and Perform the Initial Internal Zone Configuration” on page 29

Perform this task for each zone that you are installing and configuring. This section explains only the special requirements for installing Solaris Zones for use with HA for Solaris Zones. For complete information about installing and configuring Solaris Zones, see *Creating and Using Oracle Solaris Zones* , *Creating and Using Oracle Solaris 10 Zones* and *Creating and Using Oracle Solaris Kernel Zones* .

How to Enable a Zone to Run in a Failover Configuration

Before You Begin If the zone is set with `ip-type=shared` and will use a logical hostname, ensure that the `/etc/inet/netmasks` file has an IP-address subnet and netmask entry for the logical hostname. If necessary, edit the `/etc/inet/netmasks` file to add any missing entries.

1. **Register the `SUNW.HAStoragePlus` resource type.**

   ```
   # clresourcetype register SUNW.HAStoragePlus
   ```

2. **Create a failover resource group.**

   ```
   # clresourcegroup create solaris-zone-resource-group
   ```

3. **Create a resource for the zone's disk storage.**

   - **If the zone is one of the following, this step is required:**

     - A `solaris` or `solaris10` branded zone that is *not* set with the `rootzpool` zone property.
     - A kernel zone with one of the following conditions:
       - The boot device points to a `zvol`.
       - The suspend device points to a path.

     This HAStoragePlus resource is for the zonepath. The file system must be a failover file system.

     ```
     # clresource create \
     -g solaris-zone-resource-group \
     -t SUNW.HAStoragePlus \
     -p Zpools=solaris-zone-instance-zpool \
     solaris-zone-has-resource-name
     ```

   - **If the zone is one of the following, this step is optional:**

     - A kernel zone that has a storage URI that points to a logical unit or to an iSCSI device.
     - A `solaris` or `solaris10` non-global zone with both of the following conditions:
       - The `rootzpool` or `zpool` zone property is set.
       - The storage URI points to a logical unit or to an iSCSI device.

     For any other zone, this step does not apply.

a. Identify the devices to be used as boot storage and suspend storage for the kernel zone or the devices that are set for the `rootzpool` or `zpool` zone property.

```
node-1# cldev list -v d2
DID Device     Full Device Path
d2             node-1:/dev/rdsk/c0t60080E5000184744000005B4513DF1A8d0
d2             node-2:/dev/rdsk/c0t60080E5000184744000005B4513DF1A8d0

node-1# suriadm lookup-uri /dev/did/dsk/d2
dev:did/dsk/d2

node-1# cldev list -v d3
DID Device       Full Device Path
d3               node-1:/dev/rdsk/c0t60080E5000184744000005B6513DF1B2d0
d3               node-2:/dev/rdsk/c0t60080E5000184744000005B6513DF1B2d0

node-1# suriadm lookup-uri /dev/did/dsk/d3
dev:did/dsk/d3

d2 (suri=dev:did/dsk/d2) will be used for the kernel zone rpool as boot
device or for a non-global zone within the rootzpool zone property setting.

d3 (suri=dev:did/dsk/d3) will be used as suspend device or additional
delegated zpool for a non-global zone within the zpool zone property setting.
```

b. If you require device monitoring for the storage devices configured to be used by the zone, configure a `SUNW.HAStoragePlus` resource.

Specify the corresponding global device group for the `did` devices that were identified within the `GlobalDevicePaths` property in Step 32a.

i Register the `SUNW.HAStoragePlus` resource.

```
node2# clrs create -t SUNW.HAStoragePlus -g zone-rg \
-p GlobalDevicePaths=dsk/d2,dsk/d3 ha-zones-hasp-rs
```

ii Set the resource name for that `SUNW.HAStoragePlus` resource within the `HAS_RS` variable of the `sczbt_config` file.

This setting ensures that the required resource dependency gets set up for the `sczbt` component. For example:

```
HAS_RS=ha-zones-hasp-rs
```

4. **If the zone is set with `ip-type=shared` and uses a logical hostname, create a resource for that logical hostname.**

```
# clreslogicalhostname create \
-g solaris-zone-resource-group \
-h solaris-zone-logical-hostname \
solaris-zone-logical-hostname-resource-name
```

5. **Enable the failover resource group.**

```
# clresourcegroup online -eM solaris-zone-resource-group
```

How to Enable a Zone to Run in a Multiple-Masters Configuration

1. **Create a scalable resource group.**

```
# clresourcegroup create \
-p Maximum_primaries=max-number \
-p Desired_primaries=desired-number \
solaris-zone-resource-group
```

2. **Enable the scalable resource group.**

```
# clresourcegroup online -M solaris-zone-resource-group
```

How to Install a `solaris` Branded Zone and Perform the Initial Internal Zone Configuration

Perform this task on each node that is to host the `solaris` branded non-global zone. For complete information about installing a `solaris` branded non-global zone, see *Creating and Using Oracle Solaris Zones* .

Note - For procedures to install a `solaris10` branded zone or a `solaris-kz` (kernel) zone, see "How to Install a `solaris10` Branded Zone and Perform the Initial Internal Zone Configuration" on page 25 or "How to Install a `solaris-kz` Branded Zone and Perform the Initial Internal Zone Configuration" on page 29.

Before You Begin

- Consult "Planning the HA for Solaris Zones Installation and Configuration" on page 12 and then determine the following requirements for the deployment of the zone with Oracle Solaris Cluster:
  - The number of Solaris Zone instances that are to be deployed.
  - The zpool containing the file system that is to be used by each Solaris Zone instance.
- Ensure that the zone is enabled to run in a failover or multiple-masters configuration. See "How to Enable a Zone to Run in a Failover Configuration" on page 18 or "How to Enable a Zone to Run in a Multiple-Masters Configuration" on page 20.
- If the zone will run in a failover configuration and it is *not* set with the `rootzpool` zone property, ensure that the zone's zone path specifies a file system on a zpool that is managed by the `SUNW.HAStoragePlus` resource that you created in "How to Enable a Zone to Run in a Failover Configuration" on page 18.

For detailed information about configuring a `solaris` branded zone before installation of the zone, see Chapter 1, "How to Plan and Configure Non-Global Zones," in *Creating and Using Oracle Solaris Zones* .

Note - This procedure assumes you are performing it on a two-node cluster. If you perform this procedure on a cluster with more than two nodes, perform on all nodes any steps that say to perform them on both nodes.

---

1. **Assume the `root` role on one node of the cluster.**

   Alternatively, if your user account is assigned the System Administrator profile, issue commands as non-root through a profile shell, or prefix the command with the `pfexec` command.

2. **Bring the resource group online.**

   ```
   phys-schost-1# clresourcegroup online -eM solaris-zone-resource-group
   ```

3. **For zones that are *not* set with the `rootzpool` zone property, create a ZFS file-system dataset on the ZFS storage pool that you created.**

   You will use this file system as the zone root path for zone that you create later in this procedure.

   ```
   phys-schost-1# zfs create pool/filesystem
   ```

4. **Configure the zone on both nodes.**

   You must define the `osc-ha-zone` attribute in the zone configuration, setting `type` to `boolean` and `value` to `true`.

Note - If the zone is not set with the rootzpool zone property, set the zone root path to the file system that you created on the ZFS storage pool.

```
phys-schost# zonecfg -z zonename \
'create ; add attr; set name=osc-ha-zone; set type=boolean; set value=true; end;
set zonepath=/pool/filesystem/zonename ; set autoboot=false'
```

5. **Verify the zone configuration.**

```
phys-schost# zoneadm list -cv
ID NAME          STATUS        PATH                          BRAND    IP
0  global        running       /                             solaris  shared
-  zonename      configured    /pool/filesystem/zonename     solaris  shared
```

6. **Install the zone.**

   a. **(Only when rootzpool or zpool zone property is not set) Determine on which node the resource group is online.**

```
phys-schost-1# clresourcegroup status solaris-zone-resource-group
=== Cluster Resource Groups ===

Group Name                  Node Name      Suspended   Status
----------                  ---------      ---------   ------
solaris-zone-resource-group  phys-schost-1  No          Online
…
```

   Perform the rest of this step from the node that masters the resource group, or on all nodes for a multiple-master configuration.

   b. **Install the zone on each node where the resource group is online.**

```
phys-schost-N# zoneadm -z zonename install
```

   c. **Verify that the zone is installed.**

```
phys-schost-N# zoneadm list -cv
ID NAME          STATUS       PATH                          BRAND    IP
0  global        running      /                             solaris  shared
-  zonename      installed    /pool/filesystem/zonename     solaris  shared
```

   d. **Boot the zone that you created and verify that the zone is running.**

```
phys-schost-N# zoneadm -z zonename boot
phys-schost-N# zoneadm list -cv
```

```
ID NAME           STATUS       PATH                          BRAND    IP
0  global         running      /                             solaris  shared
-  zonename       running      /pool/filesystem/zonename     solaris  shared
```

e. Open a new terminal window and log in to the zone console.

Follow the interactive steps to finish the zone configuration.

```
phys-schost-N# zlogin -C zonename
```

f. Halt the zone.

The zone's status should return to `installed`.

```
phys-schost-N# zoneadm -z zonename halt
```

g. Detach the zone.

The zone state changes from `installed` to `configured`.

- **If the zone is *not* set with the `rootzpool` or `zpool` zone property, forcibly detach the zone.**

  ```
  phys-schost-N# zoneadm -z zonename detach -F
  ```

- **If the zone *is* set with the `rootzpool` or `zpool` zone property, detach the zone.**

  ```
  phys-schost-N# zoneadm -z zonename detach
  ```

7. For a failover configuration, verify that the resource group can switch over.

For a multiple-master configuration, omit this step.

a. Switch the resource group to the other node.

Input is similar to the following, where `phys-schost-1` is the node that currently masters the resource group and `phys-schost-2` is the node to which you switch the resource group.

```
phys-schost-1# clresourcegroup switch -n phys-schost-2 \
solaris-zone-resource-group
```

Note - Perform the remaining steps in this procedure from the node to which you switch the resource group, `phys-schost-2`.

---

b. Attach the zone to the node to which you switched the resource group.

```
phys-schost-2# zoneadm -z zonename attach
```

c. Verify that the zone is installed on the node.

Output is similar to the following:

```
phys-schost-2# zoneadm list -cv
ID NAME           STATUS       PATH                           BRAND      IP
0  global         running      /                              solaris    shared
-  zonename       installed    /pool/filesystem/zonename      solaris10  shared
```

d. Boot the zone.

```
phys-schost-2# zoneadm -z zonename boot
```

e. Open a new terminal window and log in to the zone.

Perform this step to verify that the zone is functional.

```
phys-schost-2# zlogin -C zonename
```

f. Halt the zone.

```
phys-schost-2# zoneadm -z zonename halt
```

g. Detach the zone.

The zone state changes from `installed` to `configured`.

- **If the zone is *not* set with the `rootzpool` or `zpool` zone property, forcibly detach the zone.**

  ```
  phys-schost-2# zoneadm -z zonename detach -F
  ```

- **If the zone *is* set with the `rootzpool` or `zpool` zone property, detach the zone.**

  ```
  phys-schost-1# zoneadm -z zonename detach
  ```

▼ How to Install a solaris10 Branded Zone and Perform the Initial Internal Zone Configuration

Perform this task on each node that is to host the solaris10 branded non-global zone. For complete information about installing a solaris10 branded zone, see *Creating and Using Oracle Solaris 10 Zones* .

Note - For procedures to install a solaris branded zone or a solaris-kz (kernel) zone, see "How to Install a solaris Branded Zone and Perform the Initial Internal Zone Configuration" on page 20 or "How to Install a solaris-kz Branded Zone and Perform the Initial Internal Zone Configuration" on page 29.

Before You Begin

- Consult "Planning the HA for Solaris Zones Installation and Configuration" on page 12 and then determine the following requirements for the deployment of the zone with Oracle Solaris Cluster:
  - The number of Solaris Zone instances that are to be deployed.
- Ensure that the zone is enabled to run in a failover or multiple-masters configuration. See "How to Enable a Zone to Run in a Failover Configuration" on page 18 or "How to Enable a Zone to Run in a Multiple-Masters Configuration" on page 20.
- If the zone will run in a failover configuration and it is *not* set with the rootzpool zone property, ensure that the zone's zone path specifies a file system on a zpool that is managed by the SUNW.HAStoragePlus resource that you created in "How to Enable a Zone to Run in a Failover Configuration" on page 18.

For detailed information about configuring a solaris10 branded zone before installation of the zone, see Chapter 4, "Configuring the solaris10 Branded Zone," in *Creating and Using Oracle Solaris 10 Zones* .

Note - This procedure assumes you are performing it on a two-node cluster. If you perform this procedure on a cluster with more than two nodes, perform on all nodes any steps that say to perform them on both nodes.

---

1. **Assume the root role on one node of the cluster.**
   Alternatively, if your user account is assigned the System Administrator profile, issue commands as non-root through a profile shell, or prefix the command with the pfexec command.

2. **Set up the system image.**
   Follow procedures in "Creating the Image for Directly Migrating Oracle Solaris 10 Systems Into Zones" in *Creating and Using Oracle Solaris 10 Zones* .

3. **For zones that are *not* set with the `rootzpool` zone property, create a ZFS file-system dataset on the ZFS storage pool that you created.**

   You will use this file system as the zone root path for the zone that you create later in this procedure.

   ```
   phys-schost-1# zfs create pool/filesystem
   ```

4. **Configure the zone on both nodes.**

   For zones that are *not* set with the `rootzpool` zone property, set the zone root path to the file system that you created on the ZFS storage pool.

   Note - You must define the `osc-ha-zone` attribute in the zone configuration, setting `type` to `boolean` and `value` to `true`.

   ```
   phys-schost# zonecfg -z zonename \
   'create ; set brand=solaris10; set zonepath=/pool/filesystem/zonename;
   add attr; set name=osc-ha-zone; set type=boolean;
   set value=true; end; set autoboot=false'
   ```

5. **Verify the zone configuration.**

   ```
   phys-schost# zoneadm list -cv
   ID NAME          STATUS        PATH                         BRAND      IP
   0 global        running       /                             solaris    shared
   - zonename      configured    /pool/filesystem/zonename     solaris10  shared
   ```

6. **For a failover configuration only, install the zone.**

   For a multiple-master configuration, omit this step.

   a. **(Only when `rootzpool` or `zpool` zone property is not set) Determine on which node the resource group is online.**

      ```
      phys-schost-1# clresourcegroup status solaris-zone-resource-group
      === Cluster Resource Groups ===

      Group Name                      Node Name          Suspended        Status
      ----------                      ---------          ---------        ------
      solaris-zone-resource-group     phys-schost-1      No               Online
      …
      ```

      Perform the rest of this step from the node that masters the resource group, or on all nodes for a multiple-master configuration.

   b. **Install the zone on each node where the resource group is online.**

```
phys-schost-N# zoneadm -z zonename install -a flarimage -u
```

c. Verify that the zone is installed.

```
phys-schost-N# zoneadm list -cv
ID NAME            STATUS       PATH                           BRAND      IP
0 global          running      /                               solaris    shared
- zonename        installed    /pool/filesystem/zonename       solaris10   shared
```

d. Boot the zone that you created and verify that the zone is running.

```
phys-schost-N# zoneadm -z zonename boot
phys-schost-N# zoneadm list -cv
ID NAME            STATUS       PATH                           BRAND      IP
0 global          running      /                               solaris     shared
- zonename        running      /pool/filesystem/zonename       solaris10   shared
```

e. Open a new terminal window and log in to the zone console.

```
phys-schost-N# zlogin -C zonename
```

Follow the interactive steps to finish the zone configuration.

f. Halt the zone.

```
phys-schost-1# zoneadm -z zonename halt
```

The zone's status should return to `installed`.

g. Detach the zone.

- **For a zone that is *not* set with the `rootzpool` or `zpool` zone property, forcibly detach the zone.**

  ```
  phys-schost-1# zoneadm -z zonename detach -F
  ```

  The zone state changes from `installed` to `configured`.

- **For a zone that is set with the `rootzpool` or `zpool` zone property, detach the zone.**

  ```
  phys-schost-1# zoneadm -z zonename detach
  ```

7. For a failover configuration, verify that the zone can switch over.

For a multiple-master configuration, omit this step.

a. Switch the resource group to the other node.

Input is similar to the following, where `phys-schost-1` is the node that currently masters the resource group and `phys-schost-2` is the node to which you switch the resource group.

```
phys-schost-1# clresourcegroup switch -n phys-schost-2 \
solaris-zone-resource-group
```

Note - Perform the remaining steps in this procedure from the node to which you switch the resource group, `phys-schost-2`.

b. Attach the zone to the node to which you switched the resource group.

```
phys-schost-2# zoneadm -z zonename attach
```

c. Verify that the zone is installed on the node.

Output is similar to the following:

```
phys-schost-2# zoneadm list -cv
ID NAME           STATUS       PATH                           BRAND      IP
0 global         running      /                              solaris     shared
- zonename       installed    /pool/filesystem/zonename      solaris10   shared
```

d. Boot the zone.

```
phys-schost-2# zoneadm -z zonename boot
```

e. Open a new terminal window and log in to the zone.

Perform this step to verify that the zone is functional.

```
phys-schost-2# zlogin -C zonename
```

f. Halt the zone.

```
phys-schost-2# zoneadm -z zonename halt
```

g. Detach the zone.

- **For a zone that is *not* set with the `rootzpool` or `zpool` zone property, forcibly detach the zone.**

  ```
  phys-schost-2# zoneadm -z zonename detach -F
  ```

The zone state changes from installed to configured.

- **For a zone that is set with the rootzpool or zpool zone property, detach the zone.**

```
phys-schost-1# zoneadm -z zonename detach
```

How to Install a solaris-kz Branded Zone and Perform the Initial Internal Zone Configuration

Perform this task on each node that is to host the solaris-kz branded zone, or kernel zone. For complete information about installing a zone, see *Creating and Using Oracle Solaris Kernel Zones* .

Note - For procedures to install a solaris branded zone or a solaris-10 branded zone, see "How to Install a solaris Branded Zone and Perform the Initial Internal Zone Configuration" on page 20 or "How to Install a solaris10 Branded Zone and Perform the Initial Internal Zone Configuration" on page 25.

Before You Begin

Consult "Planning the HA for Solaris Zones Installation and Configuration" on page 12 and then determine the following requirements for the deployment of the zone with Oracle Solaris Cluster:

- The number of Solaris Zone instances that are to be deployed.
- Ensure that the zone is enabled to run in a failover or multiple-masters configuration. See "How to Enable a Zone to Run in a Failover Configuration" on page 18 or "How to Enable a Zone to Run in a Multiple-Masters Configuration" on page 20.

Note - The storage is specified as described in the suri(5) man page. If the storage URI points to a zvol, then the corresponding zpool must be managed by a SUNW.HAStoragePlus resource. If the storage URI points to a logical unit or iSCSI device, then the SUNW.HAStoragePlus resource can be used to monitor the corresponding did device.

For detailed information about configuring a solaris-kz branded (kernel) zone before installation of the zone, see Chapter 1, "Planning and Configuring Oracle Solaris Kernel Zones," in *Creating and Using Oracle Solaris Kernel Zones* .

Note - This procedure assumes you are performing it on a two-node cluster. If you perform this procedure on a cluster with more than two nodes, perform on all nodes any steps that say to perform them on both nodes.

1. **Assume the root role on one node of the cluster.**

   Alternatively, if your user account is assigned the System Administrator profile, issue commands as non-root through a profile shell, or prefix the command with the pfexec command.

2. **Configure the zone only on the first node.**

   Observe the following requirements for the following zonecfg command:

   - Define the osc-ha-zone attribute in the zone configuration, setting type to boolean and value to true.
   - Use the did devices identified in Step 32a of this procedure.

   - **For warm migration, use the following command:**

     ```
     phys-schost-1# zonecfg -z zonename \
     'create -b; set brand=solaris-kz; add capped-memory; set physical=2G; end;
     add device; set storage=dev:did/dsk/d2; set bootpri=1; end;
     add suspend; set storage=dev:did/dsk/d3; end;
     add anet; set lower-link=auto; end; set autoboot=false;
     add attr; set name=osc-ha-zone; set type=boolean; set value=true; end;'
     ```

   - **For cold or live migration, use the following command, which omits the add suspend line:**

     ```
     phys-schost-1# zonecfg -z zonename \
     'create -b; set brand=solaris-kz; add capped-memory; set physical=2G; end;
     add device; set storage=dev:did/dsk/d2; set bootpri=1; end;
     add anet; set lower-link=auto; end; set autoboot=false;
     add attr; set name=osc-ha-zone; set type=boolean; set value=true; end;'
     ```

3. **Verify the zone configuration.**

   ```
   phys-schost-1# zoneadm list -cv
   ID NAME          STATUS        PATH                           BRAND      IP
   0 global        running       /                              solaris     shared
   - zonename      configured    /pool/filesystem/zonename      solaris-kz  shared
   ```

4. **Install the zone.**

a. Determine on which node the resource group is online.

```
phys-schost-1# clresourcegroup status solaris-zone-resource-group
=== Cluster Resource Groups ===

Group Name                      Node Name          Suspended        Status
----------                      ---------          ---------        ------
solaris-zone-resource-group     phys-schost-1      No               Online
…
```

Perform the rest of this step from the node that masters the resource group, or on all nodes for a multiple-master configuration..

b. Install the zone on each node where the resource group is online.

```
phys-schost-N# zoneadm -z zonename install
```

c. Verify that the zone is installed.

```
phys-schost-N# zoneadm list -cv
ID NAME           STATUS       PATH                           BRAND       IP
0 global          running      /                              solaris      shared
- zonename        installed    /pool/filesystem/zonename      solaris-kz   shared
```

d. Boot the zone that you created and verify that the zone is running.

```
phys-schost-N# zoneadm -z zonename boot
phys-schost-N# zoneadm list -cv
ID NAME           STATUS       PATH                           BRAND       IP
0 global          running      /                              solaris      shared
- zonename        running      /pool/filesystem/zonename      solaris-kz   shared
```

e. Open a new terminal window and log in to the zone console.

```
phys-schost-N# zlogin -C zonename
```

Follow the interactive steps to finish the zone configuration.

f. Shut down the zone.

```
phys-schost-1# zoneadm -z zonename shutdown
```

g. Forcibly detach the zone.

```
phys-schost-1# zoneadm -z zonename detach -F
```

h. Export the zone configuration on the first node, copy it to a secure location on the second node, and import the zone configuration on the second node.

This is the only supported method to copy the kernel zone configuration to another node while ensuring that it contains the encryption key for the kernel zone host data that it maintains. For more information about the kernel zone, see the solaris-kz(5) man page.

```
phys-schost-1# zonecfg -z zonename export -f /var/cluster/run/zonename.cfg
phys-schost-1# scp /var/cluster/run/zonename.cfg  root@node-2:/var/cluster/run/
phys-schost-1# rm /var/cluster/run/zonename.cfg

phys-schost-2# zonecfg -z zonename -f /var/cluster/run/zonename.cfg
phys-schost-2# rm /var/cluster/run/zonename.cfg
```

5. Switch the resource group to the other node.

Input is similar to the following, where phys-schost-1 is the node that currently masters the resource group and phys-schost-2 is the node to which you switch the resource group.

```
phys-schost-1# clresourcegroup switch -n phys-schost-2 solaris-zone-resource-group
```

Note - Perform the remaining steps in this procedure from the node to which you switch the resource group, phys-schost-2.

6. Forcibly attach the zone to the second node.

```
phys-schost-2# zoneadm -z zonename attach -x force-takeover
```

7. Verify that the zone is installed on the node.

Output is similar to the following:

```
phys-schost-2# zoneadm list -cv
ID NAME           STATUS       PATH                           BRAND       IP
0 global          running      /                              solaris     shared
- zonename        installed    /pool/filesystem/zonename      solaris-kz  shared
```

8. Boot the zone.

```
phys-schost-2# zoneadm -z zonename boot
```

9. Open a new terminal window and log in to the zone.

Perform this step to verify that the zone is functional.

```
phys-schost-2# zlogin -C zonename
```

10. (Live migration only) On both nodes, enable rad services and the kernel zone migration service.

```
phys-schost-N# svcadm enable svc:/system/rad:local svc:/system/rad:remote \
svc:/network/kz-migr:stream
```

11. (Live migration only) Enable passwordless ssh for the root user between the cluster nodes.

a. On both nodes, create the public and private ssh key for user root with an empty passphrase.

```
phys-schost-N# ssh-keygen -N '' -f /root/.ssh/id_rsa -t rsa
```

b. On each node, copy the public ssh key for user root to the other node.

Put the public key of the remote node into the authorized_keys file on the local node.

```
phys-schost-1# scp /root/.ssh/id_rsa.pub \
phys-schost-2:/var/run/phys-schost-1-root-ssh-pubkey.txt

phys-schost-2# scp /root/.ssh/id_rsa.pub \
phys-schost-1:/var/run/phys-schost-2-root-ssh-pubkey.txt

phys-schost-1# cat /var/run/phys-schost-2-root-ssh-pubkey.txt \
>> /root/.ssh/authorized_keys
phys-schost-1# rm /var/run/phys-schost-2-root-ssh-pubkey.txt

phys-schost-2# cat /var/run/phys-schost-1-root-ssh-pubkey.txt \
>> /root/.ssh/authorized_keys
phys-schost-2# rm /var/run/phys-schost-1-root-ssh-pubkey.txt
```

c. Verify that the passwordless ssh login works between each node.

Accept the public keys to continue the connection once for each node.

```
phys-schost-1# ssh root@clusternode2-priv date
…
Are you sure you want to continue connecting (yes/no)? yes

phys-schost-2# ssh root@clusternode1-priv date
…
Are you sure you want to continue connecting (yes/no)? yes
```

12. (Live migration only) Perform a live migration from the second node to the first node.

The migration is run over the cluster interconnect.

```
phys-schost-2# zoneadm -z sol-kz-fz-1 migrate ssh://clusternode1-priv
```

The zone should be running on the first node and its status on the second node should be `detached`.

13. **Shut down the zone.**

    - **For cold or warm migration, shut down the zone on `phys-schost-2`.**

      ```
      phys-schost-2# zoneadm -z zonename shutdown
      ```

    - **For live migration, shut down the zone on `phys-schost-1`.**

      ```
      phys-schost-1# zoneadm -z zonename shutdown
      ```

14. **Forcibly detach the zone.**

    - **For cold or warm migration, detach the zone from `phys-schost-2`.**

      ```
      phys-schost-2# zoneadm -z zonename detach -F
      ```

    - **For live migration, detach the zone from `phys-schost-1`.**

      ```
      phys-schost-1# zoneadm -z zonename detach -F
      ```

    The zone state changes from `installed` to `configured`.

Installing the HA for Solaris Zones Package

If you did not install the HA for Solaris Zones package during your initial Oracle Solaris Cluster installation, perform this procedure to install the package.

▼ How to Install the HA for Solaris Zones Package

Perform this procedure on each cluster node where you want the HA for Solaris Zones software to run.

1. **On the cluster node where you are installing the data service package, assume the `root` role.**

2. **Ensure that the data service package is available from the configured publisher and that the `solaris` and `ha-cluster` publishers are valid.**

```
# pkg list -a ha-cluster/data-service/ha-zones
# pkg publisher
PUBLISHER                             TYPE     STATUS   P  LOCATION
solaris                               origin   online   F  solaris-repository
ha-cluster                            origin   online   F  ha-cluster-repository
```

For information about setting the `solaris` publisher, see "Adding, Modifying, or Removing Package Publishers" in *Adding and Updating Software in Oracle Solaris 11.2*.

Tip - Use the `-nv` options whenever you install or update to see what changes will be made, such as which versions of which packages will be installed or updated and whether a new BE will be created.

If you do not get any error messages when you use the `-nv` options, run the command again without the `-n` option to actually perform the installation or update. If you do get error messages, run the command again with more `-v` options (for example, `-nvv`) or more of the package FMRI pattern to get more information to help you diagnose and fix the problem. For troubleshooting information, see Appendix A, "Troubleshooting Package Installation and Update," in *Adding and Updating Software in Oracle Solaris 11.2*.

3. **Install the HA for Solaris Zones software package.**

```
# pkg install ha-cluster/data-service/ha-zones
```

4. **Verify that the package installed successfully.**

```
$ pkg info ha-cluster/data-service/ha-zones
```

Installation is successful if output shows that `State` is `Installed`.

5. **Perform any necessary updates to the Oracle Solaris Cluster software.**

For instructions on updating your software, see Chapter 11, "Updating Your Software," in *Oracle Solaris Cluster 4.3 System Administration Guide* .

Registering and Configuring HA for Solaris Zones

Before you perform this procedure, ensure that the HA for Solaris Zones data service packages are installed.

Note - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. For Oracle Solaris Cluster Manager log-in instructions, see "How to Access Oracle Solaris Cluster Manager" in *Oracle Solaris Cluster 4.3 System Administration Guide* . After you log in, click Tasks and then click Oracle Solaris Zone to start the wizard.

This wizard requires that all cluster nodes have the same root password.

Use the configuration and registration files in the following directories to register the HA for Solaris Zones resources:

- `/opt/SUNWsczone/sczbt/util`
- `/opt/SUNWsczone/sczsh/util`
- `/opt/SUNWsczone/sczsmf/util`

The files define the dependencies that are required between the HA for Solaris Zones components. For information about these dependencies, see "Dependencies Between HA for Solaris Zones Components" on page 16.

The register script for each component reads the component configuration file and registers the resource types, `ORCL.ha-zone_sczbt`, `ORCL.ha-zone_sczsh`, and `ORCL.ha-zone_sczsmf`. The register script does not register the pure generic data service based resources. The variables for each component configuration file need to get defined as extension properties within the new resource types. The register script reads the variables from the components configuration file and relates them to the corresponding resource properties upon registration.

Registering and configuring HA for Solaris Zones involves the tasks that are explained in the following sections:

1. "Specifying Configuration Parameters for the Zone Boot Resource" on page 37
2. "Writing Scripts for the Zone Script Resource" on page 42
3. "Specifying Configuration Parameters for the Zone Script Resource" on page 43
4. "Writing a Service Probe for the Zone SMF Resource" on page 45
5. "Specifying Configuration Parameters for the Zone SMF Resource" on page 46
6. "How to Create and Enable Resources for the Zone Boot Component" on page 48
7. "How to Create and Enable Resources for the Zone Script Component" on page 56
8. "How to Create and Enable Resources for the Zone SMF Component" on page 57

Specifying Configuration Parameters for the Zone Boot Resource

HA for Solaris Zones provides the script `sczbt_register`, which automates the process of configuring the zone boot resource. By default this script obtains configuration parameters from the `sczbt_config` file in the `/opt/SUNWsczone/sczbt/util` directory. To specify configuration parameters for the zone boot resource, copy the `sczbt_config` file to a different filename and amend it as described below. It is recommended to keep this file as a future reference. The register script provides `-f` option to specify the fully qualified filename to the copied configuration file.

Each configuration parameter in the `sczbt_config` file is defined as a keyword-value pair. The `sczbt_config` file already contains the required keywords and equals signs. For more information, see "Listing of `sczbt_config`" on page 65. When you edit the `sczbt_config` file, add the required value to each keyword.

For more information, see "How to Create and Enable Resources for the Zone Boot Component" on page 48.

The keyword-value pairs in the `sczbt_config` file are as follows:

`RS=`*sczbt-rs*
`RG=`*sczbt-rg*
`PARAMETERDIR=`
`SC_NETWORK=`*true*|*false*
`SC_LH=`*sczbt-lh-rs*
`FAILOVER=true|false`
`HAS_RS=`*sczbt-has-rs*
`Zonename=`*zone-name*
`Zonebrand=`*zone-brand-type*
`Zonebootopt=`*zone-boot-options*
`Milestone=`*zone-boot-milestone*
`LXrunlevel=`*linux-runlevel*
`SLrunlevel=`*solaris-legacy-runlevel*
`Mounts=`*list-of-mountpoints*
`Migrationtype=`*cold*|*warm*|*live*

The meaning and permitted values of the keywords in the `sczbt_config` file are as follows:

`RS=`*sczbt-rs*

Specifies the name that you are assigning to the zone boot resource. You must specify a value for this keyword.

`RG=`*sczbt-rg*

Specifies the name of the resource group the zone boot resource will reside in. You must specify a value for this keyword.

`PARAMETERDIR=`

This keyword is deprecated. Leave the value empty for this keyword.

`SC_NETWORK=true|false`

Specifies whether the zone boot resource is network aware with a `SUNW.LogicalHostName` resource. You must specify a value for this keyword.

- If HA for the zone's addresses is not required, then configure the zone`s addresses by using the `zonecfg` utility.

  ```
  SC_NETWORK=false
  SC_LH=
  ```

- If only HA using a public network management (PNM) object is required, then configure the zone's addresses by using the `zonecfg` utility and then place the zone's addresses on a PNM object.

  ```
  SC_NETWORK=false
  SC_LH=
  ```

- If HA using a PNM object and protection against the failure of all physical interfaces by triggering a failover is required, choose one option from the following list:
  - If you require the `SUNW.LogicalHostName` resource type to manage one or a subset of the zone's addresses, configure a `SUNW.LogicalHostName` resource for those zone's addresses and not by using the `zonecfg` utility. Use the `zonecfg` utility to configure only the zone's addresses that are not to be under the control of the `SUNW.LogicalHostName` resource.

    `SC_NETWORK=true`
    `SC_LH=`*Name of the SUNW.LogicalHostName resource*
  - If you require the `SUNW.LogicalHostName` resource type to manage all the zone's addresses, configure a `SUNW.LogicalHostName` resource with a list of the zone's addresses and do not configure them by using the `zonecfg` utility.

    `SC_NETWORK=true`
    `SC_LH=`*Name of the SUNW.LogicalHostName resources*
  - Otherwise, configure the zone's addresses by using the `zonecfg` utility and configure a separate redundant IP address for use by a `SUNW.LogicalHostName` resource, which must not be configured using the `zonecfg` utility.

    `SC_NETWORK=false`

`SC_LH=`*Name of the SUNW.LogicalHostName resource*

`SC_LH=`*sczbt-lh-rs*

Specifies the name of the `SUNW.LogicalHostName` resource for the zone boot resource. Refer to "Requirements and Restrictions for Zone Network Addresses" on page 12 for a description of when to set this variable. This name must be the `SUNW.LogicalHostname` resource name you assigned when you created the resource in Step 4.

`FAILOVER=true|false`

Specifies whether the zone's zone path is on a highly available file system.

`HAS_RS=`*sczbt-has-rs*

Specifies the name of the `SUNW.HAStoragePlus` resource or any other cluster resource that provides highly available storage used by the zone. This name must be the resource name you assigned when you created the resource in "How to Enable a Zone to Run in a Failover Configuration" on page 18. You must specify a value for this keyword if `FAILOVER=true` is set. This will cause the `scbzt_register` script to define a `Resource_dependencies_offline_restart` resource dependency from the zone boot resource to the specified `SUNW.HAStoragePlus` resource.

The `sczbt_register` script defines a resource dependency of type `Resource_dependencies_offline_restart` as follows:

- If you set the `SC_LH` variable within the `sczbt_config` file, then the `Resource_dependencies_offline_restart` property of the `sczbt` component will contain the `SUNW.LogicalHostname` resource name as set with the `SC_LH` variable.
- If you set the `HAS_RS` variable within the `sczbt_config` file, then the `Resource_dependencies_offline_restart` property of the `sczbt` component will contain the storage resource name as set with the `HAS_RS` variable.

For a multi-master configuration, the `HAS_RS=` parameter must be empty because there is no `SUNW.HAStoragePlus` resource. All the zone paths are local to each node in that configuration.

`Zonename=`*zone-name*

Specifies the zone name. You must specify a value for this keyword.

`Zonebrand=`*zone-brand-type*

Specifies the brand type of the zone. The options that are currently supported are `solaris`, `solaris10`, and `solaris-kz`. You must specify a value for this keyword.

`Zonebootopt=`*zone-boot-options*

Specifies the zone boot option to use. Only `-s` is supported. Leaving this variable blank will cause the zone to boot to the `multi-user-server` milestone.

`Milestone=`*zone-boot-milestone*

Specifies the milestone the zone must reach to be considered successfully booted. This option is used for the `solaris`, `solaris10`, and `solaris-kz` brand type. You must specify a value for this keyword if you set the `Zonebrand` option to `solaris`, `solaris10`, or `solaris-kz`.

`LXrunlevel=`*linux-runlevel*

This option was used on Oracle Solaris 10 and is now deprecated. Any value for this keyword is ignored.

`SLrunlevel=`*solaris-legacy-runlevel*

This option was used on Oracle Solaris 10 and is now deprecated. Any value for this keyword is ignored.

`Migrationtype=`*cold*|*warm*|*live*

Specifies the type of migration that must be used for the configured kernel zone. Values for `Migrationtype` can be either cold, warm, or live. This option is only used with the `solaris-kz` brand of zone.

- With `Migrationtype=cold`, when the resource group performs a failover, the kernel zone is shut down on the current running node and freshly booted on the new node.
- With `Migrationtype=warm`, when the resource group performs a failover, the kernel zone is suspended on the current running node and booted from the suspended image on the new node.
- With `Migrationtype=live`, when the resource group performs a switchover, the kernel zone is live migrated to the new node.

`Mounts=`*list-of-mountpoints*

Specifies a space separated list of directories with their mount options, which will automatically get `lofs` mounted from the global zone into the booted zone. The mount point used in the global zone can be different to the mount point in the booted zone. Specifying a value for this keyword is optional.

The `Mounts` keyword format is as follows:

`Mounts="`*/global-zone-dir:/local-zone-dir:mount-options <next entry>*`"`

While *mount-options* can be a comma separated list of file system mount options.

The only required entry when setting this keyword is the */global-zone-dir* part of the colon separated variable. The */local-zone-dir* and *mount-options* part can be omitted.

Omitting the */local-zone-dir* part will make the zone's mount point the same as the global zone directory.

Omitting the *mount-options* part will not provide any mount options except the default options from the mount command.

Note - If you are omitting the */local-zone-dir* or the *mount-options*, you must also omit the ":" as delimiter.

Note - You must manually create any mount point directories within the booted zone that will be used within the `Mounts` keyword, before registering this resource within Oracle Solaris Cluster.

Note - If the file system of the source mount point in the global zone is mounted by a `SUNW.HAStoragePlus` resource, you must specify a strong resource dependency from the `sczbt` resource to this `SUNW.HAStoragePlus` resource.

EXAMPLE 1 Sample `sczbt_config` File

This example shows an `sczbt_config` file in which configuration parameters are set as follows:

- The name of the zone boot resource is `zone1-rs`.
- The name of the resource group for the zone boot resource is `zone1-rg`.
- Indicates that the zone's address is managed by a `SUNW.LogicalHostName` resource and is `true`.
- The name of the `SUNW.LogicalHostName` resource name for the zone boot resource is `zone1-lh`.
- Indicates that the zone boot resource's zone path is managed by a `SUNW.LogicalHostName` resource and is `true`.
- The name of the `SUNW.HAStoragePlus` resource name for the zone boot resource is `zone1-has`.
- The name of the zone is `zone1`.
- The brand type of the zone is `solaris`.
- Indicates that the zone boot resource`s boot option is `null`.
- Indicates that the zone boot resource`s milestone is `multi-user-server`.
- Defines that `/global/app/bin` from the global zone gets mounted read-only within zone `zone1` under mount point `/app/bin`.
- Defines that `/app/data` from the global zone gets mounted read-write within zone `zone1` under mount point `/app/data`.
- Defines that `/logs` from the global zone gets mounted with default mount options within zone `zone1` under mount point `/logs`.

- Defines that cold migration is performed for the zone.

```
RS=zone1-rs
RG=zone1-rg
PARAMETERDIR=
SC_NETWORK=true
SC_LH=zone1-lh
FAILOVER=true
HAS_RS=zone1-has
Zonename=zone1
Zonebrand=solaris
Zonebootopt=
Milestone=multi-user-server
Mounts="/global/app/bin:/app/bin:ro /app/data:rw /logs"
Migrationtype=cold
```

Writing Scripts for the Zone Script Resource

The zone script resource provides the ability to run commands or scripts to start, stop and probe an application within a zone. The zone script resource depends on the zone boot resource. The command or script names are passed to the zone script resource when the resource is registered and must meet with the following requirements.

- The command or script must contain the fully qualified path within the zone.
- The command or script must be executable by root.
- The command or script must return one of the following return codes.

TABLE 3 Zone Script Resource Return Codes

Return Code	Description
0	Successful completion
>0	An error has occurred
201	(Probe only) – An error has occurred that requires an immediate failover of the resource group
>0 & !=201	(Probe only) – An error has occurred that requires a resource restart

Note - For an immediate failover of the zone script resource, you must configure the resource properties `Failover_mode` and `Failover_enabled` to meet the required behavior. Refer to the `r_properties(5)` man page when setting the `Failover_mode` property and the `SUNW.gds(5)` man page when setting the `Failover_enabled` property.

EXAMPLE 2 Zone Probe Script for Apache2

This example shows a simple script to test that the Apache2 service is running, beyond the process tree existing. The script `/var/tmp/probe-apache2` must exist and being executable within the zone.

```
# cat /var/tmp/probe-apache2
#!/usr/bin/ksh
if echo "GET; exit" | mconnect -p 80 > /dev/null 2>&1
then
exit 0
else
exit 100
fi

# chmod 755 /var/tmp/probe-apache2
```

Specifying Configuration Parameters for the Zone Script Resource

HA for Solaris Zones provides the script `sczsh_register`, which automates the process of configuring zone script resource. By default this script obtains configuration parameters from the `sczsh_config` file in the `/opt/SUNWsczone/sczsh/util` directory. To specify configuration parameters for the zone script resource, copy the `sczsh_config` file to a different filename and amend it as described below. It is recommended to keep this file as a future reference. The register script provides option `-f` to specify the fully qualified filename to the copied configuration file.

Each configuration parameter in the `sczsh_config` file is defined as a keyword-value pair. The `sczsh_config` file already contains the required keywords and equals signs. For more information, see "Listing of `sczsh_config`" on page 69. When you edit the `sczsh_config` file, add the required value to each keyword.

The keyword-value pairs in the `sczsh_config` file are as follows:

`RS=`*sczsh-rs*
`RG=`*sczbt-rg*
`SCZBT_RS=`*sczbt-rs*
`PARAMETERDIR=`
`Zonename=`*sczbt-zone-name*
`ServiceStartCommand=`*sczsh-start-command*
`ServiceStopCommand=`*sczsh-stop-command*
`ServiceProbeCommand=`*sczsh-probe-command*

The meaning and permitted values of the keywords in the `sczsh_config` file are as follows:

`RS=`*sczsh-rs*

Specifies the name that you are assigning to the zone script resource. You must specify a value for this keyword.

`RG=`*sczbt-rg*

Specifies the name of the resource group the zone boot resource resides in. You must specify a value for this keyword.

`SCZBT_RS=`*sczbt-rs*

Specifies the name of the zone boot resource. You must specify a value for this keyword.

`PARAMETERDIR=`

This keyword is deprecated. Leave the value empty for this keyword.

`Zonename=`*sczbt-zone-name*

Specifies the zone name. You must specify a value for this keyword.

`ServiceStartCommand=`*sczsh-start-command*

Specifies the zone start command or script to run. You must specify a value for this keyword.

`ServiceStopCommand=`*sczsh-stop-command*

Specifies the zone stop command or script to run. You must specify a value for this keyword

`ServiceProbeCommand=`*sczsh-probe-command*

Specifies the zone probe command or script to run. You must specify a value for this keyword

EXAMPLE 3 Sample `sczsh_config` File

In this example the zone script resource uses the scripts that are available with the `pkg:/web/server/apache-22` package on Oracle Solaris 11. Before this example can be used the `Apache2` configuration file `http.conf` needs to be configured. For the purpose of this example, the delivered `/etc/apache2/2.2/http.conf` can be used. Amend the file so that you can successfully start and stop the Apache httpd server.

This example shows an `sczsh_config` file in which configuration parameters are set as follows:

- The name of the zone script resource is `zone1-script-rs`.

- The name of the resource group for the zone script resource is `zone1-rg`.
- The name of the zone boot resource is `zone1-rs`.
- The name of the zone is `zone1`.
- The name of the zone script resource start command and its parameter is `"/lib/svc/method/http-apache22 start"`.
- The name of the zone script resource stop command and its parameter is `"/lib/svc/method/http-apache22 stop"`.
- The name of the zone script resource probe command is `"/var/tmp/probe-apache2"`. This script is shown in Example 4, “Zone SMF Probe Script for Apache2,” on page 46 and must exist in `zone1`.

```
RS="zone1-script-rs"
RG="zone1-rg"
SCZBT_RS="zone1-rs"
PARAMETERDIR=
Zonename="zone1"
ServiceStartCommand="/lib/svc/method/http-apache22 start"
ServiceStopCommand="/lib/svc/method/http-apache22 stop"
ServiceProbeCommand="/var/tmp/probe-apache2"
```

Writing a Service Probe for the Zone SMF Resource

The zone SMF resource provides the ability to enable, disable, and probe an SMF service within a zone that is of brand type `solaris`, `solaris10` or `solaris-kz` . The zone SMF resource depends on the zone boot resource. Probing the SMF service is performed by running a command or script against the SMF service. The SMF service and probe command or script names are passed to the zone SMF resource when the resource is registered. The probe command or script must meet the following requirements.

- The probe command or script must contain the fully qualified path within the zone.
- The probe command or script must be executable by root.
- The probe command or script must return one of the following return codes.

TABLE 4 Zone SMF Resource Return Codes

Return Code	Description
0	Successful completion
100	An error occurred that requires a resource restart
201	An error has occurred that requires an immediate failover of the resource group

Note - For an immediate failover of the zone SMF resource, you must configure the resource properties `Failover_mode` and `Failover_enabled` to meet the required behavior. Refer to the `r_properties`(5) man page when setting the `Failover_mode` property and the `SUNW.gds`(5) man page when setting the `Failover_enabled` property.

EXAMPLE 4 Zone SMF Probe Script for Apache2

This example shows a simple script to test that the SMF Apache2 service is running, beyond the process tree existing. The script `/var/tmp/probe-apache2` must exist and being executable within the zone.

```
# cat /var/tmp/probe-apache2
# !/usr/bin/ksh
if echo "GET; exit" | mconnect -p 80 > /dev/null 2>&1
then
exit 0
else
exit 100
fi

# chmod 755 /var/tmp/probe-apache2
```

Specifying Configuration Parameters for the Zone SMF Resource

HA for Solaris Zones provides the script `sczsmf_register`, which automates the process of configuring the zone SMF resource. By default this script obtains configuration parameters from the `sczsmf_config` file in the `/opt/SUNWsczone/sczsmf/util` directory. To specify configuration parameters for the zone SMF resource, copy the `sczsmf_config` file to a different filename and amend it as described below. It is recommended to keep this file as a future reference. The register script provides option `-f` to specify the fully qualified filename to the copied configuration file.

Each configuration parameter in the `sczmf_config` file is defined as a keyword-value pair. The `sczsmf_config` file already contains the required keywords and equals signs. For more information, see “Listing of `sczsmf_config`” on page 70. When you edit the `sczsmf_config` file, add the required value to each keyword.

The keyword-value pairs in the `sczsmf_config` file are as follows:

RS=*sczsmf-rs*

```
RG=sczbt-rg
SCZBT_RS=sczbt-rs
ZONE=sczbt-zone-name
SERVICE=smf-service
RECURSIVE=true|false
STATE=true|false
SERVICE_PROBE=sczsmf-service-probe
```

The meaning and permitted values of the keywords in the `sczsmf_config` file are as follows:

`RS=`*sczsmf-rs*

Specifies the name that you are assigning to the zone SMF resource. This must be defined.

`RG=`*sczbt-rg*

Specifies the name of the resource group the zone boot resource resides in. This must be defined.

`SCZBT_RS=`*sczbt-rs*

Specifies the name of the zone boot resource. You must specify a value for this keyword.

`ZONE=`*sczbt-zone-name*

Specifies the zone name. This must be defined.

`SERVICE=`*smf-service*

Specifies the SMF service to enable/disable. This must be defined.

`RECURSIVE=true`|`false`

Specifies `true` to enable the service recursively or `false` to just enable the service and no dependents. This must be defined.

`STATE=true`|`false`

Specifies `true` to wait until the service state is reached or `false` to not wait until the service state is reached. This must be defined.

`SERVICE_PROBE=`*sczsmf-service-probe*

Specify the script to check the SMF service. Specifying a value for this keyword is optional.

EXAMPLE 5 Sample `sczsmf_config` File

In this example the zone SMF resource uses the `Apache2` SMF service that is available in Solaris 11. Before this example can be used the `Apache2` configuration file `http.conf` needs to

be configured. For the purpose of this example, the delivered `/etc/apache2/2.2/http.conf` can be used. Amend the `http.conf` file so that you can successfully start and stop the Apache httpd server.

This example shows an `sczsmf_config` file in which configuration parameters are set as follows:

- The name of the zone SMF resource is `zone1-smf-rs`.
- The name of the resource group for the zone SMF resource is `zone1-rg`.
- The name of the zone boot resource is `zone1-rs`.
- The name of the zone name is `zone1`.
- The name of the zone SMF service is `apache2`.
- Indicates that the zone SMF service Recursive option is `true`.
- Indicates that the zone SMF service State option is `true`.
- Indicates that the zone SMF service probe name is `/var/tmp/probe-apache2`. This script is shown in Example 2, “Zone Probe Script for Apache2,” on page 43 and must exist in `zone1`.

```
RS=zone1-smf-rs
RG=zone1-rg
SCZBT_RS=zone1-rs
ZONE=zone1
SERVICE=apache2
RECURSIVE=true
STATE=true
SERVICE_PROBE=/var/tmp/probe-apache2
```

▼ How to Create and Enable Resources for the Zone Boot Component

Before You Begin Ensure you have edited the `sczbt_config` file or a copy of it to specify configuration parameters for the HA for Solaris Zones zone boot component. For more information, see “Specifying Configuration Parameters for the Zone Boot Resource” on page 37.

1. **Assume the `root` role on one of the nodes in the cluster that will host the zone.**
2. **On both nodes, configure the zone-boot (`sczbt`) resource.**
   a. **Install and configure the HA for Zones agent.**

Note - Make the following adjustments if necessary:

- For a kernel zone, set `Migrationtype` to `warm` for warm migration or to `live` for live migration.
- If you are not using device monitoring, use the following settings instead for the `FAILOVER` and `HAS_RS` variables:

```
FAILOVER="false"
HAS_RS=
```

```
phys-schost# pkg install ha-cluster/data-service/ha-zones
phys-schost# cd /opt/SUNWsczone/sczbt/util
phys-schost# cp -p sczbt_config sczbt_config.zoneboot-resource
phys-schost# vi sczbt_config.zoneboot-resource
```

Add or modify the following entries in the file.

```
RS="zoneboot-resource"
RG="resourcegroup"
PARAMETERDIR=
SC_NETWORK="false"
SC_LH=""
FAILOVER="true"
HAS_RS="hasp-resource"
Zonename="zonename"
Zonebrand="brand"
Zonebootopt=""
Milestone="multi-user-server"
LXrunlevel="3"
SLrunlevel="3"
Mounts=""
Migrationtype=cold
```

Save and exit the file.

b. Configure the zone-boot resource.

The resource is configured with the parameters that you set in the zone-boot configuration file.

```
phys-schost# ./sczbt_register -f ./sczbt_config.zoneboot-resource
```

c. Verify that the zone-boot resource is enabled.

```
phys-schost# clresource enable zoneboot-resource
```

3. **Verify that the resource group can switch to another node and the ZFS storage pool successfully starts there after the switchover.**

   a. **Switch the resource group to another node.**

   ```
   phys-schost-2# clresourcegroup switch -n phys-schost-1 resourcegroup
   ```

   b. **Verify that the resource group is now online on the new node.**

   Output is similar to the following:

   ```
   phys-schost-1# clresourcegroup status
   === Cluster Resource Groups ===

   Group Name    Node Name          Suspended        Status
   ----------    ---------          ---------        ------
   resourcegroup phys-schost-1      No               Online
                 phys-schost-2      No               Offline
   ```

   c. **Verify that the zone is running on the new node.**

   ```
   phys-schost-1# zoneadm list -cv
   ID  NAME     STATUS       PATH                         BRAND      IP
   0   global   running      /                             solaris    shared
   1   zonename running      /pool/filesystem/zonename     brand      shared
   ```

Example 6 Configuring the HA for Zones Zone Boot Component for `solaris` Branded Zones

This example creates the HAStoragePlus resource `hasp-rs`, which uses a mirrored ZFS storage pool `hapool` in the resource group `zone-rg`. The storage pool is mounted on the `/hapool/solaris` file system. The `hasp-rs` resource runs on the `solaris` branded non-global zone `solariszone1`, which is configured on both `phys-schost-1` and `phys-schost-2`. The zone-boot resource `solariszone1-rs` is based on the `ORCL.ha-zone_sczbt` resource type. This example assumes that you are running the Oracle Solaris 11.3 version.

Create a resource group.

```
phys-schost-1# clresourcegroup create zone-rg
```

Create a mirrored ZFS storage pool to be used for the HA zone root path.

```
phys-schost-1# zpool create -m /ha-zones hapool mirror /dev/rdsk/c4t6d0 \
/dev/rdsk/c5t6d0
phys-schost-1# zpool export hapool
```

Create an HAStoragePlus resource that uses the resource group and mirrored ZFS storage pool that you created.

```
phys-schost-1# clresourcetype register SUNW.HAStoragePlus
```

```
phys-schost-1# clresource create -t SUNW.HAStoragePlus \
-g zone-rg -p Zpools=hapool hasp-rs
```

Bring the resource group online.

```
phys-schost-1# clresourcegroup online -eM zone-rg
```

Create a ZFS file-system dataset on the ZFS storage pool that you created.

```
phys-schost-1# zfs create hapool/solaris
```

Configure the solaris branded non-global zone.

```
phys-schost-1# zonecfg -z solariszone1 'create -b ; \
set zonepath=/hapool/solaris/solariszone1 ; add attr; \
set name=osc-ha-zone; set type=boolean; \
set value=true; end; set autoboot=false; set ip-type=shared
phys-schost-1# zoneadm list -cv
ID NAME             STATUS       PATH                           BRAND    IP
0  global           running      /                              solaris  shared
-  solariszone1     configured   /hapool/solaris/solariszone1   solaris  shared
```

Repeat on phys-schost-2.

Identify the node that masters the HAStoragePlus resource, and from that node install solariszone1.

```
phys-schost-1# clresource status
=== Cluster Resources ===

Resource Name             Node Name       Status        Message
--------------            ----------      -------        -------
hasp-rs                  phys-schost-1   Online        Online
                         phys-schost-2   Offline       Offline

phys-schost-1# zoneadm -z solariszone1 install
phys-schost-1# zoneadm list -cv
ID NAME             STATUS     PATH                           BRAND    IP
0  global           running    /                              solaris   shared
-  solariszone1     installed  /hapool/solaris/solariszone1   solaris  shared
phys-schost-1# zoneadm -z solariszone1 boot
phys-schost-1# zoneadm list -cv
ID NAME             STATUS     PATH                           BRAND    IP
0  global           running    /                              solaris  shared
-  solariszone1     running    /hapool/solaris/solariszone1   solaris  shared
```

Open a new terminal window and log in to solariszone1.

```
phys-schost-1# zoneadm -z solariszone1 halt
```

Forcibly detach the zone.

```
phys-schost-1# zoneadm -z solariszone1 detach -F
```

Switch zone-rg to phys-schost-2 and forcibly attach the zone.

```
phys-schost-1# clresourcegroup switch -n phys-schost-2 zone-rg
phys-schost-2# zoneadm -z solariszone1 attach
phys-schost-2# zoneadm list -cv
ID NAME             STATUS      PATH                           BRAND    IP
0  global           running     /                              solaris  shared
-  solariszone1     installed   /hapool/solaris/solariszone1   solaris  shared
phys-schost-2# zoneadm -z solariszone1 boot
```

Open a new terminal window and log in to solariszone1.

```
phys-schost-2# zlogin -C solariszone1
phys-schost-2# zoneadm -z solariszone1 halt
```

Forcibly detach the zone.

```
phys-schost-1# zoneadm -z solariszone1 detach -F
```

On both nodes, install and configure the HA for Zones agent.

```
phys-schost# pkg install ha-cluster/data-service/ha-zones
phys-schost# cd /opt/SUNWsczone/sczbt/util
phys-schost# cp -p sczbt_config sczbt_config.solariszone1-rs
phys-schost# vi sczbt_config.solariszone1-rs
```

On both nodes, add or modify entries in the sczbt_config.solariszone1-rs file.

```
RS="solariszone1-rs"
RG="zone-rg"
PARAMETERDIR=
SC_NETWORK="false"
SC_LH=""
FAILOVER="true"
HAS_RS="hasp-rs"
Zonename="solariszone1"
Zonebrand="solaris"
Zonebootopt=""
Milestone="multi-user-server"
LXrunlevel="3"
SLrunlevel="3"
Mounts=""
Migrationtype=cold
```

Save and exit the file.

On both nodes, configure the solariszone1-rs resource and verify that it is enabled.

```
phys-schost# ./sczbt_register -f ./sczbt_config.solariszone1-rs
phys-schost# clresource enable solariszone1-rs
```

Verify that zone-rg can switch to another node and that solariszone1 successfully starts there after the switchover.

```
phys-schost-2# clresourcegroup switch -n phys-schost-1 zone-rg
phys-schost-1# clresourcegroup status
```

```
=== Cluster Resource Groups ===

Group Name                  Node Name          Suspended        Status
----------                  ---------          ---------        ------
zone-rg                     phys-schost-1      No               Online
                            phys-schost-2      No               Offline

phys-schost-1# zoneadm list -cv
ID NAME         STATUS       PATH                           BRAND      IP
0  global       running      /                              solaris    shared
1  solariszone1 running      /hapool/solaris/solariszone1   solaris    shared
```

Example 7 Configuring the HA for Solaris Zones for a `solaris-kz` Branded Zone

This example shows how to configure a `solaris-kz` branded zone on a two-node cluster to perform warm migration.

1. Identify the devices to be used as boot storage and suspend storage for the kernel zone.

```
node-1# cldev list -v d2
DID Device     Full Device Path
d2             node-1:/dev/rdsk/c0t60080E500018474400000 05B4513DF1A8d0
d2             node-2:/dev/rdsk/c0t60080E500018474400000 05B4513DF1A8d0

node-1# suriadm lookup-uri /dev/did/dsk/d2
dev:did/dsk/d2

node-1# cldev list -v d3
DID Device       Full Device Path
d3               node-1:/dev/rdsk/c0t60080E500018474400000 05B6513DF1B2d0
d3               node-2:/dev/rdsk/c0t60080E500018474400000 05B6513DF1B2d0

node-1# suriadm lookup-uri /dev/did/dsk/d3
dev:did/dsk/d3

d2 (suri=dev:did/dsk/d2) will be used for the kernel Zone rpool as boot device

d3 (suri=dev:did/dsk/d3) will be used as suspend device
```

2. Configure the kernel zone, `sol-kz-fz1`, on node 1.

```
node-1# zonecfg -z sol-kz-fz1 \
'create -b; set brand=solaris-kz; add capped-memory; set physical=2G; end;
add device; set storage=dev:did/dsk/d2; set bootpri=1; end;
add suspend; set storage=dev:did/dsk/d3; end;
```

```
add anet; set lower-link=auto; end; set autoboot=false;
add attr; set name=osc-ha-zone; set type=boolean; set value=true; end;'
```

3. Install the kernel zone, `sol-kz-fz1`, on node 1.

```
node-1# zoneadm -z sol-kz-fz1 install
```

4. Boot the kernel zone, `sol-kz-fz1`, on node 1.

```
node-1# zoneadm -z sol-kz-fz1 boot
```

5. Perform the initial zone setup by logging on to another shell.

```
node-1# zlogin -C sol-kz-fz1
```

   Within the zone console, follow the instructions for the initial zone setup.

6. Shut down the kernel zone, `sol-kz-fz1`.

```
node-1# zoneadm -z sol-kz-fz1 shutdown
```

7. Detach the kernel zone, `sol-kz-fz1`, from node 1.

```
node-1# zoneadm -z sol-kz-fz1 detach -F
```

8. Export the kernel zone configuration on node 1, copy it to a secure location on node 2 and import the zone configuration on node 2.

   This is the only supported method to copy the kernel zone configuration to another node while ensuring that it contains the encryption key for the kernel zone host data that it maintains. For more information about kernel zones, see the `solaris-kz(5)` man page.

```
node-1# zonecfg -z sol-kz-fz1 export -f /var/cluster/run/sol-kz-fz1.cfg
node-1# scp/var/cluster/run/sol-kz-fz1.cfgroot@node-2:/var/cluster/run/
node-1# rm /var/cluster/run/sol-kz-fz1.cfg

node-2# zonecfg -z sol-kz-fz1 -f /var/cluster/run/sol-kz-fz1.cfg
node-2# rm /var/cluster/run/sol-kz-fz1.cfg
```

   Repeat this step, if you determine that it to be necessary to create a new host information encryption key, by manually using the `-x initialize-hostdata` option of the `zoneadm attach` command. Normal operation and setup of kernel zones does not require re-creating the host information.

9. Attach the kernel zone, `sol-kz-fz1`, on node 2 using the `-x force-takeover` option.

```
node-2# zoneadm -z sol-kz-fz1 attach -x force-takeover
```

10. Boot the kernel zone, `sol-kz-fz1`, on node 2.

```
node-2# zoneadm -z sol-kz-fz1 boot
on another shell:
node-2# zlogin -C sol-kf-fz1
```

11. Suspend the kernel zone, `sol-kz-fz1`, on node 2.

    ```
    node-2# zoneadm -z sol-kz-fz1 suspend
    ```

12. Detach the kernel zone, `sol-kz-fz1`, on node 2.

    ```
    node-2# zoneadm -z sol-kz-fz1 detach -F
    ```

13. Configure the failover resource group.

    ```
    node-2# clrg create zone-rg
    ```

14. **(Optional)** If you require device monitoring for the storage devices configured to be used by the kernel zone, configure a `SUNW.HAStoragePlus` resource and specify the corresponding global device group for the `did` devices identified in Step 1 within the `GlobalDevicePaths` property.

    a. Register the `SUNW.HAStoragePlus` resource type, if it is not yet registered on the cluster.

       ```
       node2# clrt register SUNW.HAStoragePlus
       ```

    b. Register the `SUNW.HAStoragePlus` resource.

       ```
       node2# clrs create -t SUNW.HAStoragePlus -g zone-rg \
       -p GlobalDevicePaths=dsk/d2,dsk/d3 ha-zones-hasp-rs
       ```

    c. Set the resource name for that `SUNW.HAStoragePlus` resource within the `HAS_RS` variable in Step 15 to ensure the required resource dependency gets setup for the `sczbt` component:

       ```
       HAS_RS=ha-zones-hasp-rs
       ```

15. Create the configuration file for the `sczbt` component to manage the kernel zone, `sol-kz-fz1`.

    ```
    node-2# vi /opt/SUNWsczone/sczbt/util/sczbt_config.sol-kz-fz1-rs
    RS=sol-kz-fz1-rs
    RG=zone-rg
    PARAMETERDIR=
    SC_NETWORK=false
    SC_LH=FAILOVER=false
    HAS_RS=
    Zonename="sol-kz-fz1"
    Zonebrand="solaris-kz"
    Zonebootopt=""
    Milestone="svc:/milestone/multi-user-server"
    LXrunlevel="3"
    SLrunlevel="3"
    Mounts=""
    Migrationtype="warm"
    ```

16. Register the `sczbt` component resource.

```
node-2# /opt/SUNWsczone/sczbt/util/sczbt_register -f \
/opt/SUNWsczone/sczbt/util/sczbt_config.sol-kz-fz1-rs
```

17. Switch the resource group online and enable the `sczbt` resource.

```
node-2# clrg online -Me zone-rg
```

    Within the zone console for the kernel zone, `sol-kz-fz1`, confirm the that the zone resumes correctly.

18. Perform switchover of the `zone-rg` resource group to node1.

```
node-2# clrg switch -n node-1 zone-rg
```

    Confirm that the kernel zone suspends on node-2 and resumes on node-1, thus performing a successful warm migration.

Next Steps Go to "Verifying the HA for Solaris Zones and Configuration" on page 57.

How to Create and Enable Resources for the Zone Script Component

Before You Begin Ensure you have edited the `sczsh_config` file or a copy of it to specify configuration parameters for the HA for Solaris Zones zone script component. For more information, see "Specifying Configuration Parameters for the Zone Script Resource" on page 43.

1. **Go to the directory that contains the script for creating the HA for Solaris Zones script resource.**

```
# cd /opt/SUNWsczone/sczsh/util
```

2. **Run the script that creates the zone script resource.**

```
# ./sczsh_register -f /mypath/sczsh_config
```

3. **Bring online the zone script resource.**

```
# clresource enable sczsh-rs
```

Next Steps Go to "Verifying the HA for Solaris Zones and Configuration" on page 57.

▼ How to Create and Enable Resources for the Zone SMF Component

Before You Begin Ensure you have edited the `sczsmf_config` file or a copy of it to specify configuration parameters for the HA for Solaris Zones zone SMF component. For more information, see "Specifying Configuration Parameters for the Zone SMF Resource" on page 46.

1. **Go to the directory that contains the script for creating the HA for Solaris Zones SMF resource.**

   ```
   # cd /opt/SUNWsczone/sczsmf/util
   ```

2. **Run the script that creates the zone SMF resource.**

   ```
   # ./sczsmf_register -f /mypath/sczsmf_config
   ```

3. **Bring online the zone SMF resource.**

   ```
   # clresource enable sczsmf-rs
   ```

Next Steps Go to "Verifying the HA for Solaris Zones and Configuration" on page 57.

Verifying the HA for Solaris Zones and Configuration

After you install, register, and configure HA for Solaris Zones, verify the HA for Solaris Zones installation and configuration. Verifying the HA for Solaris Zones installation and configuration determines if the HA for Solaris Zones data service makes your zones highly available.

▼ How to Verify the HA for Solaris Zones Installation and Configuration

1. **Assume the `root` role on a cluster node that is to host the Solaris Zones component.**

2. **Ensure all the Solaris Zone resources are online.**

   For each resource, perform the following steps.

   a. **Determine whether the resource is online.**

```
# cluster status -t rg,rs
```

b. **If the resource is not online, bring online the resource.**

```
# clresource enable solaris-zone-resource
```

3. **For a failover service configuration, switch the zone resource group to another cluster node, such as *node2*.**

```
# clresourcegroup switch -n node2 solaris-zone-resource-group
```

4. **Confirm that the resource is now online on *node2*.**

```
# cluster status -t rg,rs
```

Upgrading Non-Global Zones Managed by HA for Oracle Solaris Zones

You can upgrade Oracle Solaris non-global zones that are managed by the Oracle Solaris Cluster software. For instructions, see Chapter 2, "Upgrading Zones Managed by Oracle Solaris Cluster Software," in *Oracle Solaris Cluster 4.3 Upgrade Guide* .

Note - if you are upgrading from Oracle Solaris Cluster 4.1 version or earlier, you do not need to re-register resources that are currently using the `SUNW.gds` resource type. These resources will continue to work as they did before the upgrade.

Tuning the HA for Solaris Zones Fault Monitors

The HA for Solaris Zones fault monitors verify that the following components are running correctly:

- Zone boot resource
- Zone script resource
- Zone SMF resource

Each HA for Solaris Zones fault monitor is contained in the resource that represents Solaris Zones component. You create these resources when you register and configure HA for

Solaris Zones. For more information, see “Registering and Configuring HA for Solaris Zones” on page 35.

System properties and extension properties of these resources control the behavior of the fault monitor. The default values of these properties determine the preset behavior of the fault monitor. The preset behavior should be suitable for most Oracle Solaris Cluster installations. Therefore, you should tune the HA for Solaris Zones fault monitor *only* if you need to modify this preset behavior.

Tuning the HA for Solaris Zones fault monitors involves the following tasks:

- Setting the interval between fault monitor probes
- Setting the time-out for fault monitor probes
- Defining the criteria for persistent faults
- Specifying the failover behavior of a resource

For more information, see “Tuning Fault Monitors for Oracle Solaris Cluster Data Services” in *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide* .

Operation of the Fault Monitor for the Zone Boot Component

The fault monitor for the zone boot component ensures that the all requirements for the zone boot component to run are met:

- The corresponding `zsched` process for the zone is running.

  If this process is not running, the fault monitor restarts the zone. If this fault persists, the fault monitor fails over the resource group that contains resource for the zone boot component.
- Every logical hostname that is managed by a `SUNW.LogicalHostname` resource is operational.

  If the logical hostname is not operational, the fault monitor fails over the resource group that contains resource for the zone boot component.
- The specified milestone for the `solaris`, the `solaris10` or the `solaris-kz` zone brand type is either online or degraded.

  If the milestone is not online or degraded, the fault monitor restarts the zone. If this fault persists, the fault monitor fails over the resource group that contains resource for the zone boot component.

  To verify the state of the milestone, the fault monitor connects to the zone. If the fault monitor cannot connect to the zone, the fault monitor retries every five seconds for

approximately 60% of the probe time-out. If the attempt to connect still fails, then the fault monitor restarts the resource for the zone boot component.

Caution - The `Probe_timeout` defaults to 30 seconds. If you configure multiple Solaris HA zones on the same cluster or in combination with additional workloads, ensure that 60% of the `Probe_timeout` is enough (even under high system load) to successfully run the probe. Increase the `Probe_timeout` if the default is too sensitive in your actual deployment.

Operation of the Fault Monitor for the Zone Script Component

The fault monitor for the zone script component runs the script that you specify for the component. The value that this script returns to the fault monitor determines the action that the fault monitor performs. For more information, see Table 3, “Zone Script Resource Return Codes,” on page 42.

Operation of the Fault Monitor for the Zone SMF Component

The fault monitor for the zone SMF component verifies that the SMF service is not disabled. If the service is disabled, the fault monitor restarts the SMF service. If this fault persists, the fault monitor fails over the resource group that contains the resource for the zone SMF component.

If the service is not disabled, the fault monitor runs the SMF service probe that you can specify for the component. The value that this probe returns to the fault monitor determines the action that the fault monitor performs. For more information, see Table 4, “Zone SMF Resource Return Codes,” on page 45.

Tuning the HA for Solaris Zones `Stop_timeout` property

The HA for Solaris Zones components consist all of the resource type `SUNW.gds(5)`. As described in “Stop_command Property” in *Oracle Solaris Cluster Generic Data Service (GDS) Guide* , the value for the `Stop_timeout` should be chosen so that the `Stop_command` can successfully return within 80% of its value.

Choosing the Stop_timeout value for the Zone Boot Component

The stop method for the zone boot component spends 60% of the value for the Stop_timeout performing a complete zoneadm -z *zonename* shutdown command for the zone. If that failed, the next 20% of the value for the Stop_timeout will be spent halting the zone performing a zoneadm –z *zonename* halt command and perform some additional cleanup steps in order to force the zone into the state installed. Therefore the Stop_timeout value for the zone boot component should be computed so that 60% is enough to successfully shutdown the zone.

The default setting for the Migrationtype variable is cold. For a solaris-kz branded zone, you will be able to set the Migrationtype variable to warm or live.

If Migrationtype is set to warm, the stop method uses the "zoneadm -z *zonename* suspend" command, instead of the zoneadm -z *zonename* shutdown command. Or, if Migrationtype is set to live, the stop method uses the zoneadm -z *zonename* migrate command, instead of the zoneadm -z *zonename* shutdown command. Therefore, the Stop_timeout value must be tuned properly, when the Migrationtype variable to set to warm or live.

Choosing the Stop_timeout value for the Zone Script Component

The stop method for the zone script component calls the command or script configured for the ServiceStopCommand keyword. Therefore the Stop_timeout value for the zone script component should be computed so that 80% is enough for the configured ServiceStopCommand to succeed.

Choosing the Stop_timeout value for the Zone SMF Component

The stop method for the zone SMF component spends 60% of the value for the Stop_timeout using svcadm to disable the configured SMF service in the zone. If that failed, the next 20% of the value for the Stop_timeout will be spent to first send SIGTERM then SIGKILL to the processes associated with this SMF service. Therefore the Stop_timeout value for the zone

SMF component should be computed so that 60% is enough to successfully disable the configured SMF service in the zone.

Debugging HA for Solaris Zones

The `config` file in the `/opt/SUNWsczone/`*zone component*`/etc` directory enables you to activate debugging for Solaris Zone resources. Where *zone component* represents `sczbt` for the boot component, `sczsh` for the script component and `sczsmf` for the SMF component.

Each component of HA for Solaris Zones has a `config` that enables you to activate debugging for Solaris Zone resources. The location of this file for each component is as follows:

- For the zone boot component, this file is contained in the `/opt/SUNWsczone/sczbt/etc` directory.
- For the zone script component, this file is contained in the `/opt/SUNWsczone/sczsh/etc` directory.
- For the zone SMF component, this file is contained in the `/opt/SUNWsczone/sczsmf/etc` directory.

▼ How to Activate Debugging for HA for Solaris Zones

1. **Determine whether debugging for HA for Solaris Zones is active.**

   If debugging is inactive, `daemon.notice` is set in the file `/etc/syslog.conf`.

   ```
   # grep daemon /etc/syslog.conf
   *.err;kern.debug;daemon.notice;mail.crit/var/adm/messages
   *.alert;kern.err;daemon.erroperator
   #
   ```

2. **If debugging is inactive, edit the `/etc/syslog.conf` file to change `daemon.notice` to `daemon.debug`.**

3. **Confirm that debugging for HA for Solaris Zones is active.**

   If debugging is active, `daemon.debug` is set in the file `/etc/syslog.conf`.

   ```
   # grep daemon /etc/syslog.conf
   *.err;kern.debug;daemon.debug;mail.crit/var/adm/messages
   ```

```
*.alert;kern.err;daemon.erroperator
#
```

4. **Restart the `syslogd` daemon.**

```
# svcadm restart system-log
```

5. **Edit the `/opt/SUNWsczone/sczbt/etc/config` file to change `DEBUG=` to `DEBUG=ALL` or `DEBUG=`*sczbt-rs*.**

```
# cat /opt/SUNWsczone/sczbt/etc/config
#
# Copyright 2012 Oracle Corporation.All rights reserved.
# Use is subject to license terms.
#
# ident "@(#)config1.106/02/12"
#
# Usage:
#DEBUG=<RESOURCE_NAME> or ALL
#
DEBUG=ALL
#
```

Note - To deactivate debugging, reverse the preceding steps.

♦ ♦ ♦ APPENDIX A

Files for Configuring HA for Solaris Zones Resources

The `/opt/SUNWsczone/zone component/util` directory contains files that automate the process of configuring HA for Solaris Zones resources. Listings of these files are provided in the following sections:

Listing of `sczbt_config`

```
#
# Copyright (c) 2006, 2015, Oracle and/or its affiliates. All rights reserved.
#
#
# ident "@(#)sczbt_config 1.13 15/05/11"
#
# This file will be sourced in by sczbt_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#               RS - Name of the sczbt resource
#               RG - Name of the resource group containing the sczbt resource RS
#     PARAMETERDIR - Name of the parameter file direcrory - this variable is
#                    now deprecated and no longer used.
#       SC_NETWORK - Identfies if SUNW.LogicalHostname will be used
#   true = zone will use SUNW.LogicalHostname
#                false = zone will use it's own configuration
#
#  NOTE: If the ip-type keyword for the non-global zone is set
```

```
#        to "exclusive" or if the zone is a kernel zone, only
#        "false" is allowed for SC_NETWORK.
#
# If the ip-type keyword for the non-global zone is set to "shared",
#         the configuration of a zone's network addresses depends on
#   whether you require IPMP protection or protection against
#   the failure of all physical interfaces.
#
# If you require only IPMP protection, configure the zone's
#   addresses by using the zonecfg utility and then place the
#   zone's address in an IPMP group.
#
#    To configure this option set
#    SC_NETWORK=false
#    SC_LH=
#
# If IPMP protection is not required, just configure the
#   zone's addresses by using the zonecfg utility.
#
#    To configure this option set
#    SC_NETWORK=false
#    SC_LH=
#
# If you require protection against the failure of all physical
#   interfaces, choose one option from the following list.
#
# - If you want the SUNW.LogicalHostname resource type to manage
#     the zone's addresses, configure a SUNW.LogicalHostname
#     resource with at least one of the zone's addresses.
#
#    To configure this option set
#    SC_NETWORK=true
#    SC_LH=<Name of the SUNW.LogicalHostname resource>
#
# - Otherwise, configure the zone's addresses by using the
#     zonecfg utility and configure a redundant IP address
#     for use by a SUNW.LogicalHostname resource.
#
#    To configure this option set
#    SC_NETWORK=false
#    SC_LH=<Name of the SUNW.LogicalHostname resource>
#
# Whichever option is chosen, multiple zone addresses can be
#    used either in the zone's configuration or using several
#   SUNW.LogicalHostname resources.
#
#    e.g. SC_NETWORK=true
#    SC_LH=zone1-lh1,zone1-lh2
```

```
#
#            SC_LH - Name of the SUNW.LogicalHostname resource. If set, the
#                    sczbt_register script will list the SUNW.LogicalHostname
#                    resource within the Resource_dependencies_offline_restart
#                    property of the sczbt resource.
#         FAILOVER - Identifies if the zone's zone path is on a
#                      highly available local file system
#
#   e.g. FAILOVER=true - highly available local file system
#    FAILOVER=false - local file system
#
#           HAS_RS - Name of the SUNW.HAStoragePlus resource (or any other
#                    cluster resource providing HA storage used by the zone).
#                    If set, the sczbt_register script will list the
#                    resource within the Resource_dependencies_offline_restart
#                    property of the sczbt resource.
#

RS=
RG=
PARAMETERDIR=
SC_NETWORK=
SC_LH=
FAILOVER=
HAS_RS=

#
# The following variable will be placed in the parameter file
#
# Parameters for sczbt (Zone Boot)
#
# Zonename      Name of the zone
# Zonebrand Brand of the zone. Current supported options are
#  "native" (default), "lx", "solaris8", "solaris9", "solaris10",
#               "solaris" or "solaris-kz".
# Zonebootopt   Zone boot options ("-s" requires that Milestone=single-user)
# Milestone     SMF Milestone which needs to be online before the zone is
#  considered booted. This option is only used for the
#  "native", "solaris10", "solaris" or "solaris-kz" Zonebrand.
# LXrunlevel Runlevel which needs to get reached before the zone is
#  considered booted. This option is only used for the "lx"
#  Zonebrand.
# SLrunlevel Oracle Solaris legacy runlevel which needs to get reached
#  before the zone is considered booted. This option is only used
#  for the "solaris8" or "solaris9" Zonebrand.
# Mounts  Mounts is a list of directories and their mount options,
#  which are loopback mounted from the global zone into the
#  newly booted zone. The mountpoint in the local zone can
```

```
#  be different to the mountpoint from the global zone.
#
#  This option cannot be used with the "solaris-kz" Zonebrand.
#
#  The Mounts parameter format is as follows,
#
#  Mounts="/<global zone directory>:/<local zone directory>:<mount options>"
#
#  The following are valid examples for the "Mounts" variable
#
#  Mounts="/globalzone-dir1:/localzone-dir1:rw"
#  Mounts="/globalzone-dir1:/localzone-dir1:rw /globalzone-dir2:rw"
#
#  The only required entry is the /<global zone directory>, the
#  /<local zone directory> and <mount options> can be omitted.
#
#  Omitting /<local zone directory> will make the local zone
#  mountpoint the same as the global zone directory.
#
#  Omitting <mount options> will not provide any mount options
#  except the default options from the mount command.
#
#  Note: You must manually create any local zone mountpoint
#        directories that will be used within the Mounts variable,
#        before registering this resource within Oracle Solaris
#        Cluster.
#
# Migrationtype Defines the type of migration that should be used for a
#  configured Oracle Solaris kernel zone. Values for Migrationtype
#  can be "cold", "warm" and "live".
#
#  With Migrationtype=cold, the Oracle Solaris kernel zone is
#  shut down on the current running node and freshly booted on the
#  new node, when the resource group performs a failover or
#  switchover.
#
#  With Migrationtype=warm, the Oracle Solaris kernel zone is
#  suspended on the current running node and booted from the
#  suspended image on the new node, when the resource group
#  performs a switchover.
#
#  With Migrationtype=live, the Oracle Solaris kernel zones is live
#  migrated from the current node to the new node, when the
#  resource group performs a switchover to a new node.
#
#  If the sczbt probe indicates a problem, the Oracle Solaris
#  kernel zone is restared, similar to the case where
#  Migrationtype=cold is set.
```

```
#
#  This option is only used with the "solaris-kz" branded zone.
#

Zonename=""
Zonebrand="native"
Zonebootopt=""
Milestone="svc:/milestone/multi-user-server"
LXrunlevel="3"
SLrunlevel="3"
Mounts=""
Migrationtype="cold"
```

Listing of `sczsh_config`

```
#
# Copyright (c) 2006, 2014, Oracle and/or its affiliates. All rights reserved.
#
# ident "@(#)sczsh_config 1.5 14/04/09"
#
# This file will be sourced by sczsh_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#                  RS - Name of the resource
#                  RG - Name of the resource group containing RS
#            SCZBT_RS - Name of the SC Zone boot resource
#        PARAMETERDIR - Name of the parameter file directory - this variable is
#                       now deprecated and no longer used.
#            Zonename - Name of the zone
# ServiceStartCommand - Command including all options to start
#                        the service in the configured zone
#  ServiceStopCommand - Command including all options to stop
#                        the service in the configured zone
# ServiceProbeCommand - Command including all options to probe
#                        the service in the configured zone
#
RS=""
RG=""
SCZBT_RS=""
PARAMETERDIR=""
#
# The following parameters will be put in the agents parameter file:
```

```
#
Zonename=""
ServiceStartCommand=""
ServiceStopCommand=""
ServiceProbeCommand=""
```

Listing of sczsmf_config

```
#
# Copyright (c) 2006, 2014, Oracle and/or its affiliates. All rights reserved.
#
# ident "@(#)sczsmf_config 1.5 14/04/09"
#
# This file will be sourced in by sczsmf_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#               RS - Name of the resource
#               RG - Name of the resource group containing RS
#      SCZBT_RS - Name of the SC Zone boot resource
#          ZONE - Name of the Zone
#
# For SERVICE, RECURSIVE and STATE, refer to the svcadm(1M)
#    man page
#
#    SERVICE - {FMRI | pattern}
#    FMRI - Fault management resource identifier
#    pattern - Pattern matching a service
#
#  RECURSIVE - {false | true}  Default: true
#   False - Just enable the service and no dependents
#    True - Enable the service and recursively enable
#    its dependents
#
#  RECURSIVE=true equates to svcadm enable "-r"
#
#      STATE - {false | true} Default: true
#   False - Do not wait until service state is reached
#    True - Wait until service state is reached
#
#  STATE=true equates to svcadm enable/disable "-s"
#
```

```
#    SERVICE_PROBE - Script to check the SMF service
#
#  The optional parameter, SERVICE_PROBE, provides the
#  ability to check that the SMF service is working.
#  This must be a script within the zone and must
#  adhere to these return codes,
#
#    0 - The SMF service is working
#  100 - The SMF service should be restarted
#  201 - The SMF service should initiate a failover of
#   the Resource Group
#
#  Note: That return code 201, requires that this resource
#    has an appropriate extension property value for
#    FAILOVER_MODE and FAILOVER_ENABLED=TRUE
#
#  For FAILOVER_MODE refer to the r_properties(5) man page.
#

RS=
RG=
SCZBT_RS=
ZONE=
SERVICE=
RECURSIVE=true
STATE=true
SERVICE_PROBE=""
```

♦ ♦ ♦ APPENDIX B

HA for Solaris Zones Extension Properties

Extension properties for HA for Solaris Zones resource types are described in the following sections:

For details about system-defined properties, see the r_properties(5) man page and the rg_properties(5) man page.

ORCL.ha-zone_sczbt Extension Properties

The extension properties of the ORCL.ha-zone_sczbt resource type are as follows:

Child_mon_level

This property indicates the child monitoring level for PMF.

Default -1

Category Optional

Data Type Integer

Tunable At creation

Failover_enabled

This property determines whether to failover when retry_count is exceeded during retry_interval.

Default True

Category Optional

Data Type Boolean

Tunable When disabled

`Log_level`

This property determines the log level for the event based traces.

Default `INFO`

Category Optional

Data Type Enum

Tunable Any time

`Milestone`

This property defines the SMF milestone needed to be online to consider the zone booted.

Default `"svc:/milestone/multi-user-server"`

Category Required

Data Type String

Tunable When disabled

`Migrationtype`

This property defines the type of migration to be performed.

Default cold

Category Required

Data Type Enum {cold, live, warm}

Tunable Any time

`Monitor_retry_count`

This property indicates the number of PMF restarts allowed for the fault monitor.

Default 4

Category Optional

Data Type Integer

Tunable Any time

`Monitor_retry_interval`

This property indicates the time window, in minutes, for fault monitor restarts.

Default `2`

Category Optional

Data Type Integer

Tunable Any time

`Mounts`

This property defines a list of directories and mount options to be loopback-mounted into the non-global zone.

Default `""`

Category Optional

Data Type String

Tunable When disabled

`Network_aware`

This property determines whether the application uses a network.

Default `False`

Category Optional

Data Type Boolean

Tunable At creation

`Probe_timeout`

This property indicates the time-out value, in seconds, for the probe.

Default `30`

Category Optional

Data Type Integer

Tunable Any time

`Zonebootopt`

This property defines the zone boot option.

Default `""`

Category Optional

Data Type String

Tunable When disabled

`Zonebrand`

This property defines the zone brand type of the branded zone to be managed.

Default `solaris`

Category Required

Data Type Enum {`solaris`, `solaris10`, `solaris-kz`}

Tunable At creation

`Zonename`

This property defines the zone name of the branded zones to be managed.

Default `<unset>`

Category Required

Data Type String

Tunable When disabled

`ORCL.ha-zone_sczsh` Extension Properties

The extension properties of this resource type are as follows:

`Child_mon_level`

This property indicates the child monitoring level for PMF.

Default `-1`

Category Optional

Data Type Integer

Tunable At creation

Failover_enabled

This property determines whether to failover when `retry_count` is exceeded during `retry_interval`.

Default `False`

Category Optional

Data Type Boolean

Tunable When disabled

`Log_level`

This property determines the log level for the event based traces.

Default `INFO`

Category Optional

Data Type Enum

Tunable Any time

`Monitor_retry_count`

This property indicates the number of PMF restarts allowed for the fault monitor.

Default `4`

Category Optional

Data Type Integer

Tunable Any time

`Monitor_retry_interval`

This property indicates the time window, in minutes, for fault monitor restarts.

Default `2`

Category Optional

Data Type Integer

Tunable Any time

`Network_aware`

This property determines whether the application uses a network.

Default `False`

Category Optional

Data Type Boolean

Tunable At creation

`Probe_timeout`

This property indicates the time-out value, in seconds, for the probe.

Default `30`

Category Optional

Data Type Integer

Tunable Any time

`ServiceStartCommand`

This property defines the command including all options to start the service in the configured zone.

Default `<unset>`

Category Required

Data Type String

Tunable When disabled

`ServicStopCommand`

This property defines the command including all options to stop the service in the configured zone.

Default `<unset>`

Category Required

Data Type String

Tunable When disabled

`ServiceProbeCommand`

This property defines the command including all options to probe the service in the configured zone.

Default `<unset>`

Category Required

Data Type String

Tunable When disabled

`Zonename`

This property defines the zone name of the branded zones to be managed.

Default `<unset>`

Category Required

Data Type String

Tunable When disabled

`ORCL.ha-zone_sczsmf` Extension Properties

The extension properties of this resource type are as follows:

`Child_mon_level`

This property indicates the child monitoring level for PMF.

Default `-1`

Category Optional

Data Type Integer

Tunable At creation

Failover_enabled

This property determines whether to failover when `retry_count` is exceeded during `retry_interval`.

Default `False`

Category Optional

Data Type Boolean

Tunable When disabled

`Log_level`

This property determines the log level for the event based traces.

Default `INFO`

Category Optional

Data Type Enum

Tunable Any time

`Monitor_retry_count`

This property indicates the number of PMF restarts allowed for the fault monitor.

Default `4`

Category Optional

Data Type Integer

Tunable Any time

`Monitor_retry_interval`

This property indicates the time window, in minutes, for fault monitor restarts.

Default `2`

Category Optional

Data Type Integer

Tunable Any time

`Network_aware`

This property determines whether the application uses a network.

Default `False`

Category Optional

Data Type Boolean

Tunable At creation

`Probe_timeout`

This property indicates the time-out value, in seconds, for the probe.

Default `30`

Category Optional

Data Type Integer

Tunable Any time

`Recursive`

This property defines if just the configured SMF service or also recursively its dependents get enabled.

Default `True`

Category Optional

Data Type Boolean

Tunable When disabled

`Service`

This property defines the FMRI pattern of the SMF service to manage within the configured zone.

Default `<unset>`

Category Required

Data Type String

Tunable When disabled

`Service_probe`

This property defines the command to use for probing the configured SMF service.

Default `<unset>`

Category Required

Data Type String

Tunable When disabled

`State`

This property defines if the component needs to wait until the service state of the configured SMF service is reached or not.

Default `True`

Category Optional

Data Type Boolean

Tunable When disabled

`Zonename`

This property defines the zone name of the branded zones to be managed.

Default `<unset>`

Category	Required
Data Type	String
Tunable	When disabled

Index

T

V

Z