Application Development Framework
Application Express
Big Data
Business Intelligence
Cloud Computing
Communications
Database Performance & Availability
Data Warehousing
Database
.NET
Dynamic Scripting Languages
Embedded
Digital Experience
Enterprise Architecture
Enterprise Management
Identity & Security
Java
Linux
Mobile
Service-Oriented Architecture
Solaris
SQL & PL/SQL
Systems - All Articles
Virtualization

How to Perform System Archival and Recovery Procedures with Oracle Solaris 11

November 2011

by Jesse Butler

Feedback

How to create ZFS archives that can be used to back up and later recover an installed and configured Oracle Solaris 11 host.

The steps provided in this article can comprise the core of a basic disaster recovery plan, or they can be used to migrate a system's services to a new boot device or an entirely different system of the same model. Note that migrating the installed software to a system of a different model is not supported.

Overview of the Process

A ZFS archive is created for the root pool and its associated data sets, and also for any other ZFS pools that should be migrated or recovered, such as pools that store important third-party software or local user account data.

To back up and restore an entire system, all ZFS pools and their data sets should be archived, as described in the *Oracle Solaris Administration: ZFS File Systems* guide, along with any other non-root file systems or data required for the services that the node provides. This ensures that all configuration details, third-party software, and other node-specific elements (such as local user accounts and data) will be restored on the recovery system and that it is restored to a fully functional service state. The focus of this article is on the root pool.

Once the archive is created, it can be saved on local removable media, such as a USB drive, or sent across the network to a file server from which it can later be retrieved. When it is time to make use of the archive, the following procedure is used:

The recovery system is booted from the Oracle Solaris 11 installation media and a superuser-privileged shell is started.
A boot disk device is selected and configured and the new ZFS root pool is created.
The archived ZFS data sets are recovered to the new pool.
Final configuration is performed, and the system is then rebooted.

Requirements and Caveats

Any host running Oracle Solaris 11 is a candidate for this procedure. For the system archive to be restored to a new disk or system, the following requirements must be met:

The archived system and recovery system must be the same model and must meet the Oracle Solaris 11 minimum requirements.
The disk(s) that will house the new ZFS pool(s) must be at least as large in total capacity as the space allocated in the archived pool(s) (more details are provided below).
Root access is required on both the archived system and the recovery system.
Note that the archives will carry with them all software and configuration information that resides in the ZFS data sets that are archived. This includes, but is not limited to, the following:

The operating system and related configuration and tunings
All boot environments (BEs) and previous ZFS snapshots
Network configuration, including host name, routing information, and name services configuration
All locally installed software and related data that is locally stored
Locally configured user accounts and local user data
Zones and related configuration data
This means that in most scenarios, after the steps outlined below have been completed, no additional configuration operations will be required.

No hardware-specific configuration data is carried in the archive image. If you are using this procedure to move to an entirely new system of the same model, hardware-specific system characteristics that will not transfer with the backup include, but are not limited to, the following:

Disk capacity and configuration (including ZFS pool configurations)
Memory capacity and configuration
Hardware Ethernet address
Installed hardware peripherals
Of specific note regarding installed hardware peripherals, if the system to be recovered makes use of directly connected external storage or special networking hardware, such as Fibre Channel or InfiniBand adapters, those will need to be installed on the recovery system to access the storage.

Phase 1: Archive Creation

This section describes preparation that needs to be done and the creation of the archive.

Preparation

To prepare for recovery, note the disk topology and ZFS pool configuration for each pool that will be archived. Again, for the purposes of this article, the focus is on the root pool. The target disks on the recovery system will need to be configured similarly and the new ZFS pools will need to be sized appropriately. At a minimum, the allocated amount of each pool (the `ALLOC` column in the `zpool list` output shown below) is required to ensure there is enough room to restore the data sets on the recovery system.

```
# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool      68G  51.6G  16.4G  75%  1.00x  ONLINE  -
```

If any pool's capacity (as shown in the `CAP` column) exceeds 80%, best practices dictate that the pool should be grown to plan for capacity. Increasing the headroom in the pool might also be beneficial for performance, depending upon other configuration elements and the workload. For more information about managing ZFS file systems and related performance, refer to *Oracle Solaris Administration: ZFS File Systems*.

To prepare for later restoration, the outputs of various commands should be saved to a file that is kept with the archive for reference during recovery. The commands shown in Listing 1 are suggested as a bare minimum, but other configuration information might be useful, depending upon the system configuration. The commands shown in Listing 1 with example output are for the root pool (`rpool`) only.

Listing 1: Commands Whose Output Should Be Saved

```
# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool      68G  51.6G  16.4G  75%  1.00x  ONLINE  -

# zpool get all rpool
NAME   PROPERTY       VALUE                 SOURCE
rpool  size           68G                   -
rpool  capacity       75%                   -
rpool  altroot        -                     default
rpool  health         ONLINE                -
rpool  guid           18397928369184079239  -
rpool  version        33                    default
rpool  bootfs         rpool/ROOT/snv_175a   local
rpool  delegation     on                    default
rpool  autoreplace    off                   default
rpool  cachefile      -                     default
rpool  failmode       wait                  default
rpool  listsnapshots  off                   default
rpool  autoexpand     off                   default
rpool  dedupditto     0                     default
rpool  dedupratio     1.00x                 -
rpool  free           16.4G                 -
rpool  allocated      51.6G                 -
rpool  readonly       off                   -

# zpool status
  pool: rpool
 state: ONLINE
  scan: none requested
config:

                NAME        STATE     READ WRITE CKSUM
                rpool       ONLINE       0     0     0
                c5t0d0s0    ONLINE       0     0     0

errors: No known data errors

# format c5t0d0s0
selecting c5t0d0s0
[disk formatted]
/dev/dsk/c5t0d0s0 is part of active ZFS pool rpool. Please see zpool(1M).


FORMAT MENU:
                disk       - select a disk
                type       - select (define) a disk type
                partition  - select (define) a partition table
                current    - describe the current disk
                format     - format and analyze the disk
                repair     - repair a defective sector
                label      - write label to the disk
                analyze    - surface analysis
                defect     - defect list management
                backup     - search for backup labels
                verify     - read and display labels
                save       - save new disk/partition definitions
                inquiry    - show disk ID
                volname    - set 8-character volume name
                !<cmd>     - execute <cmd>, then return
                quit
format> p

PARTITION MENU:
                0      - change `0' partition
                1      - change `1' partition
                2      - change `2' partition
                3      - change `3' partition
                4      - change `4' partition
                5      - change `5' partition
                6      - change `6' partition
                7      - change `7' partition
                select - select a predefined table
                modify - modify a predefined partition table
                name   - name the current table
                print  - display the current table
                label  - write partition map and label to the disk
                !<cmd> - execute <cmd>, then return
                quit
partition> p
Current partition table (original):
Total disk cylinders available: 14087 + 2 (reserved cylinders)

        Part      Tag    Flag     Cylinders         Size            Blocks
          0       root    wm       1 - 14086       68.35GB    (14086/0/0) 143339136
          1 unassigned    wm       0                0         (0/0/0)             0
          2     backup    wu       0 - 14086       68.35GB    (14087/0/0) 143349312
          3 unassigned    wm       0                0         (0/0/0)             0
          4 unassigned    wm       0                0         (0/0/0)             0
          5 unassigned    wm       0                0         (0/0/0)             0
          6 unassigned    wm       0                0         (0/0/0)             0
          7 unassigned    wm       0                0         (0/0/0)             0

partition> ^D
#
```

The information above from the system being archived, along with anything else that might be useful during recovery, should be placed in a file that is stored alongside the archive files for use later during recovery.

Alternatively, you can use the Oracle Explorer Data Collector to gather all system configuration information for later reference. Oracle Explorer Data Collector and its related documentation can be found at My Oracle Support (support contract and login required).

For additional information about ZFS administration and capacity planning, refer to *Oracle Solaris Administration: ZFS File Systems*.

Archive Creation

To archive the root pool and include all snapshots and BEs, a ZFS replication stream is created. First, you must create a recursive snapshot from the top level of the pool. In the same way, you can archive other pools that need to be archived and carried over to a recovered host.

Note that `rpool` is the default root pool name, but the root pool might be named differently on any given system. If all pools and data sets are archived, this isn't overly important. However, if only the root file system is needed or only a select set of BEs are required for the backup, and it is unclear on which pool they reside, this can be determined by using the command `beadm list -d`. From this point onward, the default name `rpool` is used to reference the root pool.

The following command creates a recursive snapshot named `archive` of the root pool. You can select a snapshot name based upon the date or whatever descriptive labels you desire.

```
# zfs snapshot -r rpool@archive
```

The recursive snapshot has now been created, but the swap and dump device snapshots should be deleted from it, because they likely do not contain any data relevant to system migration or recovery. Also, removing them typically reduces the size of the archive significantly.

The following commands delete the default-named swap and dump device snapshots, though there might be more deployed on a host.

Note: Regarding the dump device, while it is possible that the dump device has data that has not yet been extracted to the `/var` data set (in the form of a core archive), it is unlikely. If this is the case, and the contents of the dump device should be preserved, the contents should be dumped out to the file system prior to deleting the dump device snapshot. See `dumpadm`(1M) for further details.

```
# zfs destroy rpool/swap@archive
# zfs destroy rpool/dump@archive
```

To determine if more than the default-named devices are in place, use `swap`(1m) and `dumpadm`(1m) to list the names of swap and dump devices, respectively.

The snapshot is now prepared. The next step is to send it to a file for archival. In the event that more than one ZFS pool is being archived, each pool will have a snapshot, and each snapshot will be sent to its own archive file. Again, for the purposes of this example, the following steps focus on creating the archive for the root pool. However, any other pools on the system can be archived in the same manner.

The ZFS `send` command is piped into a `gzip` command, which results in a compressed file that contains the pool snapshot's archive. When creating this archive file, it is a good idea to use some unique naming scheme that reflects the host name, the date, or other descriptive terms that will be useful in determining the contents of the archive at a later date.

The archive file can be saved locally for later relocation or created on removable media. Note that although compression is being used, enough storage space should be available either locally on or the file server for the archives. A good rule of thumb is to have enough capacity for the sum of the `ALLOC` amounts reported by `zpool list`.

To create the archive file locally, use the following command. The recovery image file name can be any string that helps identify this archive for later use. A common choice is to use the host name plus the date, for example.

```
# zfs send -Rv rpool@archive | gzip > /path/to/archive_$(hostname)_$(date +%Y%m%d).zfs.gz
```

The archive file should now be moved to a file server for later retrieval.

Optionally, the archive file can be written directly to an NFS-mounted path, as shown below.

```
# zfs send -Rv rpool@archive | gzip > /net/FILESERVER/path/to/archive_$(hostname)_$(date +%Y%m%d).zfs.gz
```

Similarly, the archive file can be streamed to a file server using `ssh`.

```
# zfs send -Rv rpool@archive | gzip | ssh USER@FILESEVER "cat> /path/to/archive_$(hostname)_$(date +%Y%m%d).zfs.gz"
```

Note that while streaming the archive across the network with this last option, the `ssh` transfer does not support any sort of suspend and resume functionality. Therefore, if the network connection is interrupted, the entire command will need to be restarted.

It is highly recommended that the archive files be stored on a file system that is backed up.

Now that the recovery archive has been created, the local snapshots can be destroyed.

```
# zfs destroy -r rpool@archive
```

Phase 2: Recovery from the Archive

System Boot

The recovery phase can occur whenever system recovery or system migration operations need to be done.

To get started, the recovery system needs to be booted from the Oracle Solaris 11 installation media. The install media used to boot the recovery system should be the same version of Oracle Solaris 11 that the archive was built from. For example, if the archive was created on an Oracle Solaris 11 11/11 host, the Oracle Solaris 11 11/11 installation media should be used in this phase. The system can be booted from DVD, USB device, or the network. Note that the system is not installed from this media, but rather, the media is used only to boot the system. Once booted, a shell will be started where the recovery procedure can begin.

To boot from DVD or USB install media, insert the media and select the appropriate device as the boot device. With a LiveCD, the GNOME desktop session can be used for the recovery procedure; once the desktop has started, start a terminal window to carry out the rest of this procedure. If text-based media is used, choose to exit to a shell when the Text Installer menu comes up.

The Oracle Solaris Automated Installer (AI) or a local copy of AI media can also be used to boot a system. On an x86 host, selecting the "Text Installer and command line" GRUB menu entry will run the Text Installer, where a shell can be selected from the menu.

Similarly, on a SPARC host, booting the AI media (either locally or over the network) without starting an automated install will allow you to select the shell from the Text Installer menu. This can be done by invoking the following boot command at the boot prompt on SPARC machines:

```
{0} ok boot net:dhcp
```

For further information on booting systems with Oracle Solaris 11, refer to the x86 and SPARC booting information in *System Administration Guide: Basic Administration*.

Boot Device and Root Pool Preparation

The first step is to configure the new boot disk device. See the *Oracle Solaris Administration: Devices and File Systems* guide and the x86 and SPARC booting information in *System Administration Guide: Basic Administration* for information on how to manage disk devices, how to determine the boot device, and how to change the default boot device, if necessary.

Note that, as previously referenced, the original disk layout can be replicated or a different one can be used, as long as the following steps are taken and space at the beginning of the disk is reserved for boot data. In addition, it is not necessary for the root pool (or other recovery target pools) to be the same size as the original one. It is, however, necessary that the new pools be large enough to contain all data in the respective archive file (that is, as large as the `ALLOC` section in the `zpool list` output, as described earlier).

Once the boot device is selected, it is configured as required, based upon the initial disk configuration on the archived system. To reiterate, what is required is that ultimately the ZFS pools that are created are large enough to store the data sets, as described by the `ALLOC` amounts shown in the output of `zpool list`.

The `format`(1M) command is used to configure the disk partitions and/or slices as desired. For boot devices, a VTOC label should be used, and the default configuration is a full-device slice 0 starting at cylinder 1. The files that were saved as part of the archive creation can provide guidance on how to best set up the boot device.

To begin, select the desired boot device from the `format` utility's menu, as shown in Listing 2.

Listing 2: Selecting the Boot Device

```
# format
Searching for disks...done

c3t3d0: configured with capacity of 68.35GB
```

```
AVAILABLE DISK SELECTIONS:
      0. c3t2d0 <SEAGATE-ST973401LSUN72G-0556 cyl 8921 alt 2 hd 255 sec 63>
         /pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@2,0
      1. c3t3d0 <FUJITSU-MAY2073RCSUN72G-0401 cyl 14087 alt 2 hd 24 sec 424>
         /pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@3,0
Specify disk (enter its number): 1
selecting c3t3d0
[disk formatted]
```

With the disk selected, on x86 systems, you might need to create an `fdisk` partition.

```
format> fdisk
No fdisk table exists. The default partition for the disk is:

 a 100% "SOLARIS System" partition

Type "y" to accept the default partition,  otherwise type "n" to edit the
partition table.
y
format>
```

After this is done, you can configure the slices as needed. Listing 3 shows an example of setting up a full-capacity (or "all hog") slice 0, which is the default configuration. The slice starts at cylinder 1 to leave room for boot software at the beginning of the disk. Note that the partition table might look different based upon system architecture, disk geometry, and other variables.

Listing 3: Example of Setting Up a Full-Capacity Slice

```
format> partition
partition> print
Current partition table (default):
Total disk cylinders available: 8921 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
 0 unassigned    wm       0               0         (0/0/0)            0
 1 unassigned    wm       0               0         (0/0/0)            0
 2     backup    wu       0 - 8920       68.34GB    (8921/0/0) 143315865
 3 unassigned    wm       0               0         (0/0/0)            0
 4 unassigned    wm       0               0         (0/0/0)            0
 5 unassigned    wm       0               0         (0/0/0)            0
 6 unassigned    wm       0               0         (0/0/0)            0
 7 unassigned    wm       0               0         (0/0/0)            0
 8       boot    wu       0 -    0        7.84MB    (1/0/0)        16065
 9 unassigned    wm       0               0         (0/0/0)            0

partition> 0
Part      Tag    Flag     Cylinders        Size            Blocks
 0 unassigned    wm       0               0         (0/0/0)            0

Enter partition id tag[unassigned]: root
Enter partition permission flags[wm]:
Enter new starting cyl[1]: 1
Enter partition size[0b, 0c, 1e, 0.00mb, 0.00gb]: $
partition>
```

After the slices are configured as needed, label the disk, as shown in Listing 4. The overall layout should then be confirmed prior to moving on to the next step.

Listing 4: Labeling the Disk

```
partition> label
Ready to label disk, continue?
y
partition> print
Current partition table (unnamed):
Total disk cylinders available: 8921 + 2 (reserved cylinders)

        Part      Tag    Flag     Cylinders        Size            Blocks
         0       root    wm       1 - 8920       68.33GB    (8920/0/0) 143299800
         1 unassigned    wm       0               0         (0/0/0)            0
         2     backup    wu       0 - 8920       68.34GB    (8921/0/0) 143315865
         3 unassigned    wm       0               0         (0/0/0)            0
         4 unassigned    wm       0               0         (0/0/0)            0
         5 unassigned    wm       0               0         (0/0/0)            0
         6 unassigned    wm       0               0         (0/0/0)            0
         7 unassigned    wm       0               0         (0/0/0)            0
         8       boot    wu       0 -    0        7.84MB    (1/0/0)        16065
         9 unassigned    wm       0               0         (0/0/0)            0

partition> ^D
```

For more information regarding managing disk devices, see the *Oracle Solaris Administration: Devices and File Systems* guide.

ZFS Pool Creation and Archive Restoration

With the disk configured, the new root pool can be created on slice 0 by using the following command:

```
# zpool create rpool cXtXdXs0
```

Note that if the archived system's root pool did not use the default name `rpool`, its name should be used here instead. Although the recovery procedure will complete successfully with a differently named pool, the resulting ZFS file system might take on a different mountpoint, which might lead to confusion. Therefore, the recovery ZFS pools should be created with the same names as the archived ZFS pools.

Any other ZFS pools that are needed for archive restoration can be created at this time. Note that if an existing pool on the system already uses the desired name, you will need to select another name. For more information on ZFS pool creation, see the *Oracle Solaris Administration: ZFS File Systems* guide.

The next step is to restore the ZFS data sets from the archive file(s). If the archive is stored on removable media, that media should be attached and configured now so the file(s) can be accessed. For information on configuring removable media, please see the *Oracle Solaris Administration: Devices and File Systems* guide.

Once the file is accessible locally, the data sets can be restored by using the following command.

```
# gzcat /path/to/archive_myhost_20111011.zfs.gz | zfs receive -vF rpool
```

If the files are stored on a networked file server, the following command can be used to stream the archive and restore the data sets.

```
# ssh USER@FILESERVER "cat /path/to/archive_myhost_20111011.zfs.gz" | gzip -d | zfs receive -vF rpool
```

If other pools were archived for restoration on this host, they can be restored at this point with the same ZFS operation shown above. For additional information on recovering ZFS data sets, see *Oracle Solaris Administration: ZFS File Systems*.

The data restoration portion of the procedure is now complete. Some final steps now must be performed to ensure that the recovered system will boot as expected.

Phase 3: Configuration and Validation

First, swap and dump devices must be created for use with the recovered system. Note that the default-named devices are being used here and, as such, no further administrative tasks are required (for example, adding the swap device using the `swap(1m)` command) since they were already in use and are configured to run with this system at boot time. If the target system has a memory configuration that varies from the system that was archived, the swap and dump devices might require a different size, but the names are still the same as for the previous configuration and, thus, they will be configured properly on the first boot of the recovered system.

The swap and dump devices should be sized according to the advice in the *Oracle Solaris Administration: Devices and File Systems* and *Oracle Solaris Administration: ZFS File Systems* guides, which is roughly as follows.

Table 1. Sizes for swap and dump Devices

Physical Memory	Swap size	Dump size
System with up to 4 GB of physical memory	1 GB	2 GB
Mid-range server with 4 GB to 8 GB of physical memory	2 GB	4 GB
High-end server with 16 GB to 32 GB of physical memory	4 GB	8 GB+
System with more than 32 GB of physical memory	1/4 total memory size	1/2 total memory size

Note that after the system is booted, additional swap devices can be added, if needed. Please see the documentation referenced above for further information regarding management of these devices.

To recreate swap and dump devices with appropriate capacities, use the following commands. Note that in this example, the recovery system has 8 GB memory installed.

```
# zfs create -b $(pagesize) -V 2GB rpool/swap
# zfs set primarycache=metadata rpool/swap
# zfs create -b 128k -V 4GB rpool/dump
```

Next, the desired BE must be configured using the `beadm(1M)` command. The `beadm list` command displays a list of all available BEs.

```
# beadm list
BE        Active Mountpoint Space  Policy Created
--        ------ ---------- -----  ------ -------
solaris-2 -      -          2.02G  static 2011-09-23 10:00
solaris   -      -          16.56M static 2011-09-22 21:51
```

To work with the desired BE's root file system, it must be mounted. To do this, use `beadm mount`, as follows, which uses the `solaris-2` BE as an example.

```
# beadm mount solaris-2 /tmp/mnt
```

The BE's root file system can now be accessed through the `/tmp/mnt` mountpoint. The first step is to install the boot software, which allows the host to boot the new root pool. The steps are different depending upon architecture, as shown below. Both examples use the `/tmp/mnt` BE mountpoint, as shown above.

On an x86-based host:

```
# installgrub /tmp/mnt/boot/grub/stage1 /tmp/mnt/boot/grub/stage2 /dev/rdsk/cXtXdXs0
```

On a SPARC-based host:

```
# installboot -F zfs /tmp/mnt/usr/platform/`uname -i`/lib/fs/zfs/bootblk /dev/rdsk/cXtXdXs2
```

It is possible that the same devices will not be in use, or they will be configured in a different manner, on the new system. Therefore, clear the device file system using the following command.

```
# devfsadm -Cn -r /tmp/mnt
```

Now we will direct the system to perform a reconfiguration boot on first boot. This will configure any new device hardware (as related to the archival system versus the recovery system). A reconfiguration boot is forced by placing a file named `reconfigure` at the top level of the BE's root file system. This function is not persistent because the file is removed and, thus, the reconfiguration occurs only on the first boot after the file is placed.

To set up a reconfiguration boot, create this file in the active BE's mounted file system. The file system can then be unmounted using the `beadm unmount` command.

```
# touch /tmp/mnt/reconfigure
# beadm unmount solaris-2
```

Finally, the desired BE needs to be activated using the `beadm activate` command.

```
# beadm activate solaris-2
```

Once activated, the `Active` column of the BE in the `beadm list` output show `R`, which indicates active on reboot. This can be confirmed by a second invocation of `beadm list`. For additional information regarding BE management, see *Creating and Administering Oracle Solaris Boot Environments*.

The system can now be rebooted. The system should be as the archived system was, barring any changes in physical topology, peripheral devices, or other hardware-related changes. Software configuration and data should be carried over from the archived system, as well as any secondary ZFS pools that were recovered, as discussed above. Network configuration should be examined and verified. If the system has been recovered on a new network, or if network-related configuration elements have changed since the archive was created, some modifications might be required. Please see the Oracle Solaris Administration guides for further information regarding configuration.

Conclusion

This article demonstrated a set of procedures by which an installed and configured Oracle Solaris 11 host can be archived and restored. These procedures can be used as part of an overall disaster recovery plan, or they can be used to migrate business services hosted on an Oracle Solaris 11 system either to another boot device or to an entirely different system of the same model.

For More Information

Here are some additional resources:

Feedback

Download Oracle Solaris 11
Access Oracle Solaris 11 product documentation
Access all Oracle Solaris 11 how-to guides
Learn more with Oracle Solaris 11 training and support

Revision 1.0, 11/04/2011

E-mail this page Printer View

ORACLE CLOUD

Learn About Oracle Cloud Computing

Get a Free Trial

Learn About DaaS

Learn About SaaS

Learn About PaaS

Learn About IaaS

Learn About Private Cloud

Learn About Managed Cloud

JAVA

Learn About Java

Download Java for Consumers

Download Java for Developers

Java Resources for Developers

Java Cloud Service

Java Magazine

CUSTOMERS AND EVENTS

Explore and Read Customer Stories

All Oracle Events

Oracle OpenWorld

JavaOne

COMMUNITIES

Blogs

Discussion Forums

Wikis

Oracle ACEs

User Groups

Social Media Channels

SERVICES AND STORE

Log In to My Oracle Support

Training and Certification

Become a Partner

Find a Partner Solution

Purchase from the Oracle Store

CONTACT AND CHAT

Phone: +1.800.633.0738

Global Contacts

Oracle Support

Partner Support

Subscribe Careers Contact Us Site Maps Legal Notices Terms of Use Privacy Cookie Preferences Oracle Mobile