Oracle Solaris 11 Cheat Sheet
General Administration

Installation

Automated Installer (AI) is the new network based multi-client provisioning system on Oracle Solaris 11. AI provides hands-free installation of both SPARC and x86 systems by using an installation service that installs systems from software package repositories on the network.

Create an install service from a downloaded ISO file, specifying x86 based DHCP client starting at address 192.168.1.210 with a total count of 10 addresses:

```
# installadm create-service -n s11x86 -i 192.168.1.210 -c 10 -s
/path/to/solaris-11-1111-ai-x86.iso
```

List all enabled services:

```
# installadm list
```

List any installation manifests associated with the install services:

```
# installadm list -m
```

Export the default installation manifest associated with the `s11x86` service:

```
# installadm export -n s11x86 -m orig_default > manifest.xml
```

Import a manifest to be associated with the `s11x86` service:

```
# installadm update-manifest -n s11x86 -m orig_default \
-f manifest.xml
```

List any system configuration profiles associated with the install services:

```
# installadm list -p
```

Create a system configuration profile interactively, saving the contents to a file:

```
# sysconfig create-profile -o profile.xml
```

Validate a system configuration profile against the default x86 install service:

```
# installadm validate -n default-i386 -P profile.xml
```

Associate a system configuration profile with the deafult x86 install service and give it a name `sc-profile`:

```
# installadm create-profile -n default-i386 -f profile.xml \
-p sc-profile
```

Apply a criteria that all clients must have 4096MB memory or greater to the manifest `s11manifest` of `s11x86` service:

```
# installadm set-criteria -m s11manifest -n s11x86 \
-a MEM="4096-unbounded"
```

System Configuration

Common system configuration tasks have changed in Oracle Solaris 11 with the Service Management Facility (SMF) configuration repository being used to store configuration data. With the addition of configuration layers, administrators now have better control and assurance that their configuration changes will be preserved across system updates.

Configuring nodename:

```
# svccfg –s svc:/system/identity:node setprop \
config/nodename = “myhost”
# svcadm refresh svc:/system/identity:node
# svcadm restart svc:/system/identity:node
```

Configuring console keyboard layout:

```
# svccfg –s keymap:default setprop keymap/layout = UK-English
# svcadm refresh keymap
# svcadm restart keymap
```

Configuring system locale:

```
# svccfg –s timezone:default setprop \
timezone/localtime = astring: US/Mountain
# svcadm refresh timezone:default
```

Unconfigure a system and start an interactive configuration tool on reboot:

```
# sysconfig configure -s
```

Create a system configuration profile:

```
# sysconfig create-profile -o sc-profile.xml
```

Configure a system according to a system configuration profile:

```
# sysconfig configure -c sc-profile.xml
```

Did you know?

You can find out more information about Oracle Solaris 11 including full product documentation, how to guides, and other cheat sheets on Oracle Technology Network: http://www.oracle.com/technetwork/server-storage/solaris11/overview/index.html

Oracle Solaris 11 Administrator's Cheat Sheet

Users and Groups

The traditional `root` account has been changed to a 'root' role on all Oracle Solaris 11 installations as part of the Role Based Access Control (RBAC) feature set. This change gives improved auditability across the operating system, and the ability for administrators to delegate various system tasks to others in a safe way.

Revert to `root` as normal user account:

```
# rolemod –K type=normal root
```

Configure `root` as a role (default):

```
# usermod –K type=role root
```

Add a new user and delegate him the `System Adminstrator` profile:

```
# useradd -d /export/home/joerg -P "System Administrator" joerg
```

Boot Environments

Boot Environments are individual bootable instances of the operating system that take advantage of the Oracle Solaris ZFS filesystem snapshot and clone capability. During a system update, new boot environments are created so that system software updates can be applied in a safe environment. Should anything go awry, administrators can boot back into an older boot environment. Boot environments have low overhead and can be quickly created giving administrators an ideal best practice for any system maintenance work.

Create a boot environment:

```
# beadm create solaris-05032012
```

Activate a boot environment:

```
# beadm activate solaris-05032012
```

Delete a boot environment:

```
# beadm destroy solaris-05032012
```

Show boot environments from SPARC boot PROM:

```
ok boot -L
```

Boot into a boot environment from SPARC boot PROM:

```
ok boot -Z rpool/ROOT/solaris-05032012
```

Packaging

Oracle Solaris 11 includes IPS, a new network-centric package management framework with automatic dependency checking. IPS has integrated package and patching, and can seamlessly manage system updates to Oracle Solaris Zones environments.

Install a package called `diagnostic/wireshark`:

```
# pkg install diagnostic/wireshark
```

Install a group package to provide a desktop environment:

```
# pkg install solaris-desktop
```

Update all possible packages to the newest version, including any zones:

```
# pkg update
```

Do a dry run of a system update to understand what packages may change:

```
# pkg update -nv
```

Uninstall a package called `diagnostic/wireshark`:

```
# pkg uninstall wireshark
```

List all packages installed on a system:

```
# pkg list
```

Get more information about an installed package called `diagnostic/wireshark`:

```
# pkg info wireshark
```

List the contents of an installed package called `diagnostic/wireshark`:

```
# pkg contents wireshark
```

Search all packages in the configured repositories for a file called `math.h`:

```
# pkg search math.h
```

Search for all packages installed on a system that have a dependency on `library/libxml2`:

```
# pkg search -l -o pkg.name 'depend::library/libxml2'
```

List currently associated package publishers:

```
# pkg publisher
```

Connect to the Oracle support repository and update the system:

```
# pkg set-publisher -g https://pkg.oracle.com/solaris/support \
-G http://pkg.oracle.com/solaris/release -k /path/to/ssl_key \
-c /path/to/ssl_cert solaris
# pkg update
```

Oracle Solaris 11 Administrator's Cheat Sheet

File systems – Basic ZFS Administration

Oracle Solaris ZFS is the default root file system on Oracle Solaris 11. ZFS has integrated volume management, preserves the highest levels of data integrity and includes a wide variety of data services such as data deduplication, RAID and data encryption.

Create a ZFS pool with a single disk:

```
# zpool create testpool c3t2d0
```

Create a ZFS pool with 3 disks in RAID0 configuration:

```
# zpool create testpool c3t2d0 c3t3d0 c3t4d0
```

Create a ZFS pool with 3 disks in RAID1 configuration:

```
# zpool create testpool mirror c3t2d0 c3t3d0 c3t4d0
```

Create a ZFS pool with 3 disks in a RAIDZ configuration (single parity):

```
# zpool create testpool raidz c2t2d0 c3t3d0 c3t4d0
```

Create a ZFS pool with 1 disk and 1 disk as seperate ZIL (ZFS Intent Log):

```
# zpool create testpool c3t2d0 log c3t3d0
```

Create a ZFS pool with 1 disk and 1 disk as L2ARC (Level 2 storage cache):

```
# zpool create testpool c3t2d0 cache c3t3d0
```

Share a filesystem via NFS:

```
# zfs create zpool/fs1
# zfs set share=name=fs1,path=/rpool/fs1,prot=nfs rpool/fs1
# zfs set sharenfs=on rpool/fs1
```

Share a filesystem via CIFS:

```
# pkg install service/filesystem/smb
# svcadm enable -r smb/server
# echo "other password required pam_smb_passwd.so.1 nowarn" \
>> /etc/pam.conf
# smbadm enable-user joerg
# zfs set share=name=sh1,path=/rpool/fs1,prot=smb rpool/fs1
# zfs set sharesmb=on rpool/fs1
```

Use shadow migration:

```
# pkg install shadow-migration
# svcadm enable shadowd
# zfs set readonly=on path/to/data
# zfs create -o shadoow=file:///path/to/data \
target/new/path/to/data
```

Disk Devices

Show all disks on a system:

```
# cfgadm -s "select=type(disk)"
```

Configure a disk to be used via iSCSI

```
# svcadm enable svc:/network/iscsi/initiator
# iscsiadm modify initiator-node -A myclient
# iscsiadm add discovery-address 10.211.55.200
# iscsiadm discovery -t enable
# devfsadm -c iscsi
```

Replace a faulty disk `c1t1d0` from ZFS pool `testpool`:

```
# zpool offline testpool c1t1d0
# cfgadm -c unconfigure c1::dsk/c1t1d0
# cfgadm -c configure c1::dsk/c1t1d0
# zpool replace testpool c1t1d0
# zpool online testpool c1t1d0
```

Mirror existing boot disk c3t0d0s0 with disk c3t2d0s0

```
# fdisk -B c3t2d0s0
# prvtoc /dev/rdsk/c3t0d0s0 | fmthard -s - /dev/rdsk/c3t2d0s0
```

On x86 systems:

```
# installgrub /boot/grub/stage1 /boot/grub/stage2 \
/dev/rdsk/c3t2d0s0
```

On SPARC systems:

```
# installboot -F zfs \
/usr/platform/`uname -i`/lib/fs/zfs/bootblk/dev/rdsk/c3t2d0s0
```

Oracle Solaris 11 Administrator's Cheat Sheet

Oracle Solaris Zones

Oracle Solaris Zones provide isolated and secure virtual environments running on a single operating system instance, ideal for application deployment. When administrators create a zone, an application execution environment is produced in which processes are isolated from the rest of the system.

Create a zone with an exclusive IP network stack:

```
# zonecfg -z testzone
testzone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:testzone> create
zonecfg:testzone> set zonepath=/zones/testzone
zonecfg:testzone> set autoboot=true
zonecfg:testzone> verify
zonecfg:testzone> commit
zonecfg:testzone> exit
```

List all running zones verbosely:

```
# zoneadm list -v
```

List all configured zones:

```
# zoneadm list -c
```

List all installed zones:

```
# zoneadm list -i
```

Install a zone:

```
# zoneadm -z testzone install
```

Boot a zone:

```
# zoneadm -z testzone boot
```

List configuration about a zone:

```
# zoneadm -z testzone list
```

Login to a zone:

```
# zlogin -C testzone
```

Halt a zone

```
# zoneadm -z testzone halt
```

Shutdown a zone

```
# zoneadm -z testzone shutdown
```

Monitor a zone for CPU, memory and network utilization every 10 seconds:

```
# zonestat -z testzone 10
```

Service Management Facility

Service Management Facility (SMF) provides a framework for managing services on Oracle Solaris including the ability to automatically restart any service after failure. Each service instance is named with a fault management resource indicator (FMRI).

Show all services (including disabled services):

```
# svcs
```

List detailed information about `system/zones`:

```
# svcs -l system/zones
```

List processes associated with the `network/netcfg` service:

```
# svcs -p network/netcfg
```

Show why services that are enabled but are not running, or preventing other services from running:

```
# svcs -xv
```

Enable a service called `network/dns/client`:

```
# svcadm enable network/dns/client
```

Restart a service called network/nfs/server using an abbreviated FMRI:

```
# svcadm restart nfs/server
```

Disable a service called `network/ssh`:

```
# svcadm disable network/ssh
```

Display all properties and values in the SMF configuration repository for the service `network/ssh`:

```
# svcprop network/ssh
```

Interactively display the `general/enabled` property within the SMF configuration repository for the service `network/ssh`:

```
# svccfg
svc:> select ssh:default
svc:/network/ssh:default> listprop general/enabled
svc:/network/ssh:default> exit
```

Set the port number of the application/pkg/server service to 10000:

```
# svccfg -s application/pkg/server setprop pkg/port=10000
# svcadm refresh application/pkg/server
```

Configure email notifications for all services that drop from `online` to `maintenance` state:

```
# svccfg setnotify -g from-online,to-maintenance \
mailto:admin@myhost.org
```

List all configuration changes that have been made in the SMF configuration repository to the `name-service/switch` service:

```
# svccfg -s name-service/switch listcust -L
```

Oracle Solaris 11 Administrator's Cheat Sheet

Networking

Oracle Solaris 11 uses profile based networking configuration, comprised of two configuration modes - manual and automatic. These modes differ in how administrators configure the system, either manually using `dladm` and `ipadm`, or through creating and applying network configuration profiles.

Networking — Manual Administration

Switch to manual network configuration:

```
# netadm enable –p ncp defaultfixed
```

Show physical network interfaces:

```
# dladm show-phys
```

Create interface with static IPv4 configuration:

```
# ipadm create-ip net0
# ipadm create-addr –T static –a local=10.9.8.7/24 net0/addr
# ipadm show-addr
```

Create interface with DHCP configuration:

```
# ipadm create-ip net0
# ipadm create-addr –T dhcp net0/addr
```

Create interface with auto-generated IPv6 configuration:

```
# ipadm create-ip net0
# ipadm create-addr –T addrconf net0/addr
```

Configure default route:

```
# route –p add default 192.168.1.1
```

Activate DNS configuration:

```
# svccfg –s dns/client setprop config/nameserver = \
net_address: 192.168.1.1
# svccfg –s dns/client setprop config/domain = \
astring: “myhost.org”
# svccfg –s name-service/switch setprop config/host = \
astring: \“files dns\”
# svcadm refresh name-service/switch
# svcadm refresh dns/client
```

Activate DNS configuration (alternate approach by editing `/etc/resolv.conf` and `/etc/nsswitch.conf` and then importing these modifications into SMF)

```
# nscfg import –f svc:/system/name-service/switch:default
# nscfg import –f svc:/network/dns/client:default
# svcadm refresh dns/client
```

Networking – Automatic Administration

Create a network configuration profile:

```
# netcfg create ncp datacenter
# netcfg
netcfg> select ncp datacenter
netcfg:ncp:datacenter> create ncu phys net0
Created ncu ‘net0’. Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
ipv4-addsrc (dhcp) [dhcp|static]> static
ipv4-addr> 192.168.1.27
ipv4-default-route> 192.168.1.1
netcfg:ncp:datacenter:ncu:net0> end
Committed changes
netcfg:ncp:datacenter> exit
```

Create a network location profile:

```
# netcfg
netcfg> create loc datacenter
Created loc 'datacenter'. Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-
all]> conditional-any
conditions> ip-address is 192.168.1.27
nameservices (dns) [dns|files|nis|ldap] dns
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]> manual
dns-nameservice-domain> datacenter.myhost.org
dns-nameservice-servers> 192.168.1.1
dns-nameservice-search>
dns-nameservice-sortlist>
dns-nameservice-options>
nfsv4-domain>
ipfilter-config-file>
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file>
ipsecpolicy-config-file>
netcfg:loc:datacenter>
netcfg:loc:datacenter> exit
Committed changes
```

Activate a network configuration profile:

```
# netadm enable -p ncp datacenter
```

Networking – Advanced Administration

Create a virtual network interface over existing physical interface `net0` with address 192.168.0.80:

```
# dladm create-vnic -l net0 vnic0
# ipadm create-ip vnic0
# ipadm create-addr -T static -a 192.168.0.80 vnic0/v4
```

Create two virtual network interfaces over a virtual switch (without a physical network interface):

```
# dladm create-etherstub stub0
# dladm create-vnic -l stub0 vnic0
# dladm create-vnic -l stub0 vnic1
```

Reduce the bandwidth of the virtual network interface `vnic0` to 100Mbps:

```
# dladm set-linkprop -p maxbw=100 vnic0
```

Restrict the bandwidth going to IP address 192.168.0.30 by creating a flow on virtual network interface `vnic0`, then restrict its bandwidth to 50Mbps:

```
# flowadm add-flow -l vnic0 -a remote_ip=192.168.0.30 flow0
# flowadm set-flowprop -p maxbw=50 flow0
```

Restrict network traffic to TCP for a local port 443 for network interface `net0`:

```
# flowadm add-flow -l net0 -a transport=TCP,local_port=433 flow0
```

Activating Jumbo Frames (ethernet packets greater than 1500 bytes):

```
# dladm set-linkprop -p mtu=9000 net0
```

Configure Link Aggregation:

```
# dladm create-aggr -l net0 -l net1 aggr0
# ipadm create-ip aggr0
# ipadm create-addr -T static -a 10.1.1.2/24 aggr0/v4
```

Configure VLANS:

```
# dladm create-vlan -l net0 -v 100 administration1
# dladm create-vlan -l net0 -v 2 production1
# ipadm create-ip administration1
# ipadm create-ip production1
# ipadm create-addr -T static -a 192.168.2.2/24 \
administration1/v4static
# ipadm create-addr -T static -a 192.168.1.2/24 \
production1/v4static
```

Networking – Advanced Administration (Continued),

Configure an IPMP group:

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ip net2
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i net0 -i net1 -i net2 ipmp0
# ipadm create-addr -T static -a 192.168.1.27/24 ipmp0/v4
# ipadm create-addr -T static -a 192.168.1.50/24 net0/test
# ipadm create-addr -T static -a 192.168.1.51/24 net1/test
# ipadm create-addr -T static -a 192.168.1.52/24 net2/test
```

Oracle Solaris 11 Administrator's Cheat Sheet

Contact Us

This Oracle Solaris 11 Administrator's Cheat Sheet was written by Joerg Moellenkamp, Senior Account Architect for Oracle. Joerg writes a blog that can be found at http://www.c0t0d0s0.org/.

For more information about Oracle Solaris 11, visit oracle.com/solaris or call +1.800.ORACLE1 to speak to an Oracle representative. Last updated: July 17, 2012.

Oracle is committed to developing practices and products that help protect the environment

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

Hardware and Software, Engineered to Work Together